目 录

第一章 概 述

1.1 偏微分方程工具箱的功能

偏微分方程工具箱（PDE Toolbox）提供了研究和求解空间二维偏微分方程问题的一个强大而又灵活实用的环境。PDE Toolbox 的功能包括：

(1) 设置 PDE（偏微分方程）定解问题，即设置二维定解区域、边界条件以及方程的形式和系数；

(2) 用有限元法（FEM）求解 PDE，即网格的生成、方程的离散以及求出数值解；

(3) 解的可视化。

无论是高级研究人员还是初学者，在使用 PDE Toolbox 时都会感到非常方便。只要 PDE 定解问题的提法正确，那么，启动 MATLAB 后，在 MATLAB 工作空间的命令行中键入 pdetool，系统立即产生偏微分方程工具箱（PDE Toolbox）的图形用户界面（Graphical User Interface，简记为 GUI），即 PDE 解的图形环境，这时就可以在它上面画出定解区域、设置方程和边界条件、作网格剖分、求解、作图等工作，详见 1.4 节中的例子。我们将在第二章详细介绍 GUI 的使用，在第三章给出大量典型例子和应用实例。除了用 GUI 求解 PDE 外，也可以用M文件的编程计算更为复杂的问题，详见第三章和第四章的内容。

1.2 PDE Toolbox 求解的问题及其背景

1.2.1 方程类型

PDE Toolbox 求解的基本方程有椭圆型方程、抛物型方程、双曲型方程、特征值方程、椭圆型方程组以及非线性椭圆型方程。

椭圆型方程：

$$-\nabla\cdot(c\nabla u)+au=f,\ \text{in}\ \Omega,$$

其中Ω是平面有界区域，c, a, f以及未知函数 u 是定义在Ω上的实(或复)函数。

抛物型方程：

$$d\frac{\partial u}{\partial t}-\nabla\cdot(c\nabla u)+au=f,\ \text{in }\Omega.$$

双曲型方程：

$$d\frac{\partial^2 u}{\partial t^2}-\nabla\cdot(c\nabla u)+au=f,\ \text{in }\Omega.$$

特征值方程：

$$-\nabla\cdot(c\nabla u)+au=\lambda du,\ \text{in }\Omega,$$

其中 d 是定义在Ω上的复函数，λ是待求的特征值。在抛物型方程和双曲型方程中，系数 c, a, f 和 d 可以依赖于时间 t。

可以求解非线性椭圆型方程：

$$-\nabla\cdot(c(u)\nabla u)+a(u)u=f(u),\ \text{in }\Omega,$$

其中 c, a 和 f 可以是解 u 的函数。还可以求解如下 PDE 方程组：

$$\begin{cases}-\nabla\cdot(c_{11}\nabla u_1)-\nabla\cdot(c_{12}\nabla u_2)+a_{11}u_1+a_{12}u_2=f_1,\\-\nabla\cdot(c_{21}\nabla u_1)-\nabla\cdot(c_{22}\nabla u_2)+a_{21}u_1+a_{22}u_2=f_2.\end{cases}$$

利用命令行可以求解高阶方程组。对于椭圆型方程，可以用自适应网格算法，还能与非线性解结合起来使用。

另外，对于 Poisson 方程还有一个矩形网格的快速求解器。

1.2.2 边界条件

(1) Dirichlet 条件

$$hu=r.$$

(2) Neumann 条件

$$\boldsymbol{n}\cdot(c\nabla u)+qu=g,$$

其中$\boldsymbol{n}$是$\partial\Omega$上的单位外法向矢量，g, q, h 和 r 是定义在 $\partial\Omega$ 上的函数。对于特征值问题仅限于齐次条件：$g=0$，$r=0$。对于非线性情形，系数 g,q,h 和 r 可以依赖于 u；对于抛物型方程和双曲型方程，系数可以依赖于时间 t。对于方程组情形，Dirichlet 边界条件为

$$h_{11}u_1+h_{12}u_2=r_1,\ \ h_{21}u_1+h_{22}u_2=r_2.$$

而一般的 Neumann 条件为

$$\boldsymbol{n}\cdot(c_{11}\nabla u_1)+\boldsymbol{n}\cdot(c_{12}\nabla u_2)+q_{11}u_1+q_{12}u_2=g_1,$$

$$\boldsymbol{n}\cdot(c_{21}\nabla u_1)+\boldsymbol{n}\cdot(c_{22}\nabla u_2)+q_{21}u_1+q_{22}u_2=g_2.$$

混合边界条件为

$$h_{11}u_1 + h_{12}u_2 = r_1,$$

$$\boldsymbol{n}\cdot(c_{11}\nabla u_1) + \boldsymbol{n}\cdot(c_{12}\nabla u_2) + q_{11}u_1 + q_{12}u_2 = g_1 + h_{11}\mu,$$

$$\boldsymbol{n}\cdot(c_{21}\nabla u_1) + \boldsymbol{n}\cdot(c_{22}\nabla u_2) + q_{21}u_1 + q_{22}u_2 = g_2 + h_{12}\mu.$$

其中μ的计算要使得 Dirichlet 条件满足。在有限元法中，Dirichlet 条件也称为本质边界条件，Neumann 条件也称为自然边界条件，关于有限元法详见第五章。

1.2.3 PDE 模型的背景

Toolbox 中所解的 PDE 模型有着广泛的背景，它来自工程和科学的许多分支，现举例如下：

- 椭圆型和抛物型方程来自定常和非定常传输问题
- 多孔介质的流动和扩散问题
- 绝缘和导体材料的静电场问题
- 势流
- 双曲型方程来自暂态和谐波在声音和电磁场中的传播
- 薄膜的横振动
- 特征值问题来自例如求解薄膜和结构力学的固有振动问题

Toolbox 对于偏微分方程和偏微分方程数值解（特别是有限元法）的教学也大有好处。

1.3 如何使用 PDE Toolbox

1.3.1 定解问题的设置

最简单的办法是在 PDE Tool 上直接使用图形用户界面（GUI）。设置定解问题包括三个模式（Mode）：

(1) Draw 模式：使用 CSG（几何结构实体模型）对话框画几何区域，包括矩形、圆、椭圆和多边形，也可以将它们组合使用。

(2) Boundary 模式：在各个边界段上给出边界条件。

(3) PDE 模式：确定方程的类型、系数 c, a, f 和 d。也能够在不同子区域上设置不同的系数（反映材料的性质）。

1.3.2 解 PDE 问题

用 GUI 解 PDE 问题主要使用下面两个模式：

(1) Mesh 模式：生成网格，自动控制网格参数。

(2) Solve 模式：对于椭圆型方程还能求非线性和自适应解。对于抛物型和双曲型方程，设置初始边值条件后能求出给定 t 时刻的解。对于特征值问题，能求出给定区间内的特征值。求解后可以加密网格再求解。

1.3.3 使用 Toolbox 求解非标准的问题

对于非标准的问题，可以用 PDE Toolbox 的函数，或者用 FEM（有限元法）求解更为复杂的问题。见第四章关于命令函数部分。

1.3.4 计算结果的可视化

从 GUI 能够使用 Plot 模式实现可视化。可以使用 Color, Height 和 Vector 等作图。对于抛物型和双曲型方程，还可以生成解的动画。这些操作通过命令行都很容易实现。

1.3.5 应用领域

在应用界面提供了如下应用领域：

- 结构力学——平面应力问题
- 结构力学——平面应变问题
- 静电场问题
- 静磁场问题
- 交流电磁场问题
- 直流导体介质问题
- 热传导问题
- 扩散问题

这些界面都有对话框，它包括 PDE 的系数、边界条件、解的性质等。许多例子不但有 GUI 的使用方法，还有命令行的说明。

1.4 解偏微分方程的一个例子

解 Poisson 方程 $-\Delta u = f$，边界条件为齐次 Dirichlet 类型。

第一步：启动 MATLAB，键入 pdetool，按回车键确定便可启动 GUI，然后在 Options 菜单下选择 Grid 命令，打开栅格　栅格的使用，能使用户容易确定所绘图形的大小，如图 1-1。

第二步：分步完成平面几何造型：R1-C1-E1+R2+C2。用菜单或快捷工具，分别画矩形 R1、矩形 R2、椭圆 E1、圆 C1、圆 C2。画圆时，首先选中椭圆工

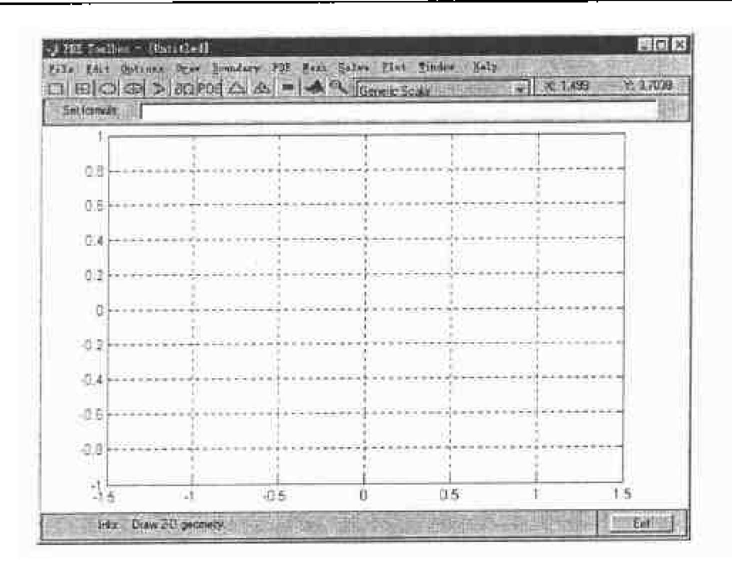


图 1-1

具，按鼠标右键并拖动即可，或者在按 Ctrl 的同时，拖动鼠标也可绘制圆。然后在 Set formula 栏，进行编辑并用算术运算符将图形对象名称连接起来，或删出默认的表达式直接键入 R1-C1-E1+R2+C2，如图 1-2。若需要，还可进行储存，形成 M 文件。

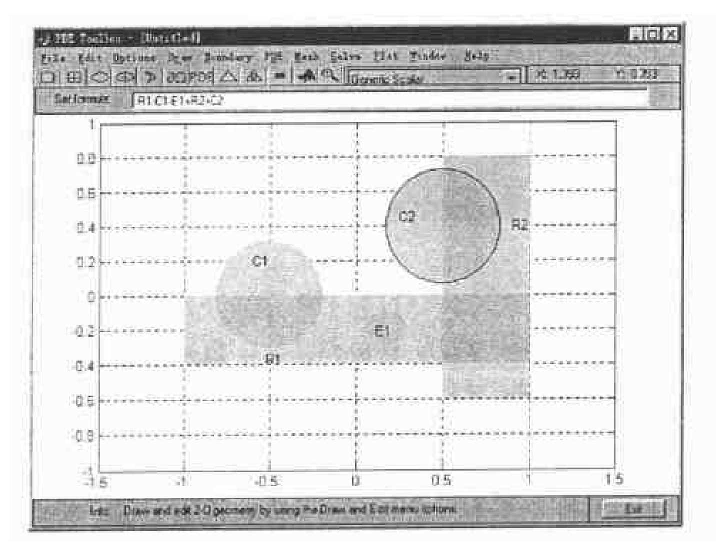


图 1-2

选择 Boundary 菜单中 Boundary Mode 命令，进入边界模式。单击 Boundary 菜单中 Remove All Subdomain Borders 选项，去除子域边界，如图 1-3。如果想将几何信息和边界信息进行存储，应选择 Boundary 菜单中的 Export Decomposed Geometry, Boundary Cond's...命令，将它们分别储存于 g,b 变量中，通过 MATLAB 形成 M 文件。

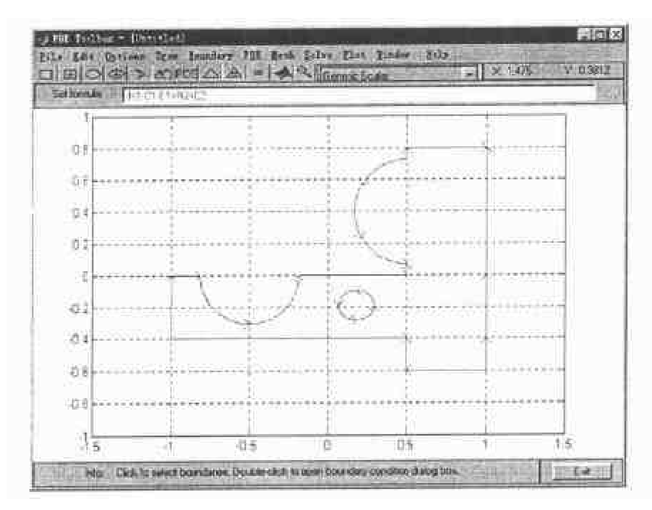


图 1-3

第二步：选取边界，单击 Boundary 菜单中 Specify Boundary Conditions...选项，打开 Boundary Conditions 对话框，输入边界条件，如图 1-4。本例取缺省条件，即将全部边界设为齐次 Dirichlet 条件，边界颜色显示为红色。

图 1-4

第四步：选择 PDE 菜单中 PDE Mode 命令，进入 PDE 模式。单击 PDE 菜单中 PDE Specification...选项，打开 PDE Specification 对话框，设置方程类型。本例取缺省设置，类型为椭圆型，参数 c,a,f 分别为 1,0,10，如图 1-5。

图 1-5

第五步：选择 Mesh 菜单中 Initialize Mesh 命令，进行网格剖分，如图 1-6。

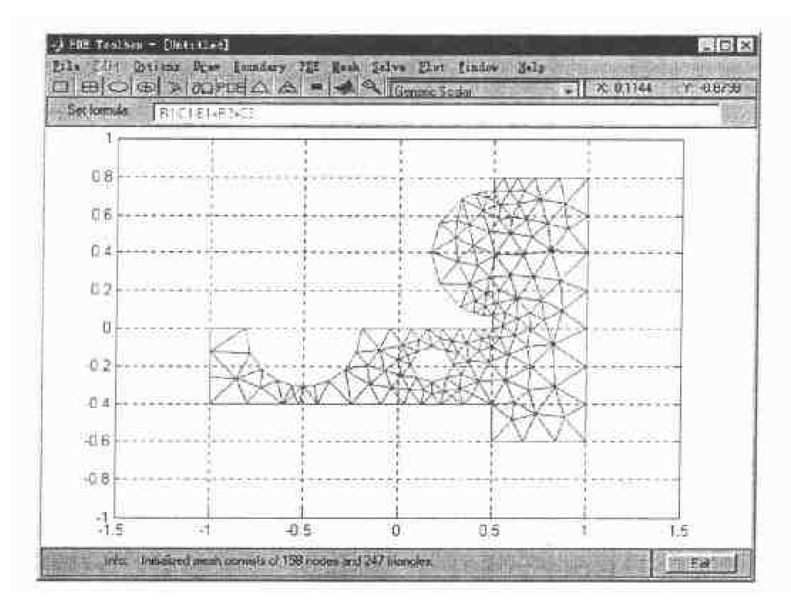


图 1-6

第六步：选择 Mesh 菜单中 Refine Mesh 命令，对网格加密，如图 1-7。

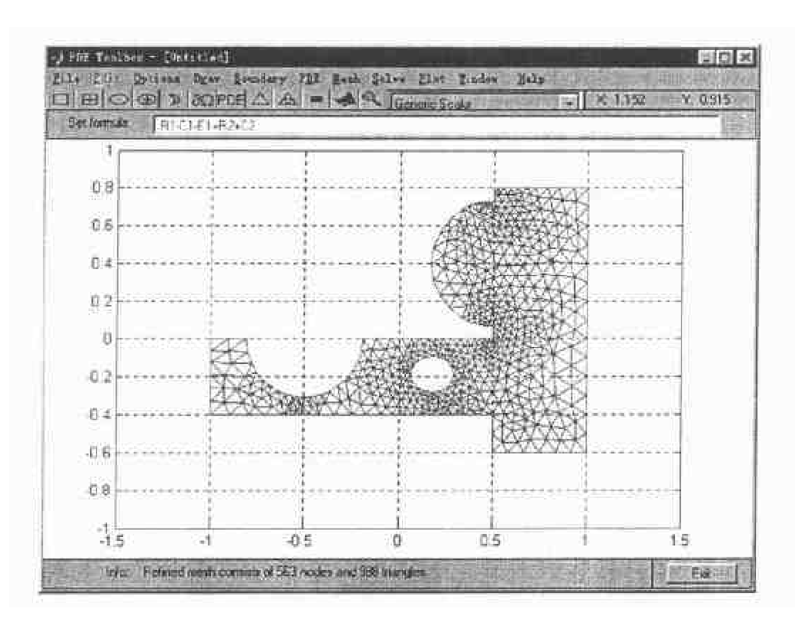


图 1-7

第七步：选择 Solve 菜单中 Solve PDE 命令，解偏微分方程并显示图形解，如图 1-8。

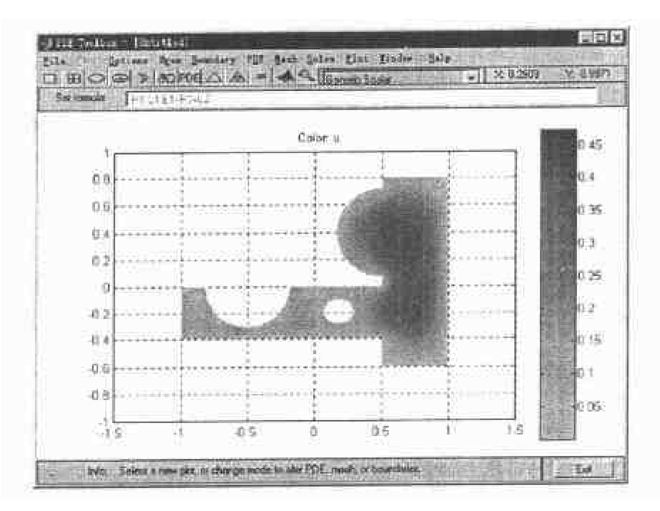


图 1-8

第八步：单击 Plot 菜单中 Parameters...选项，打开 Plot Selection 对话框，选中 Color, Height (3-D plot)和 Show mesh 三项，如图 1-9。然后单击 Plot 按钮，显示三维图形解，如图 1-10。

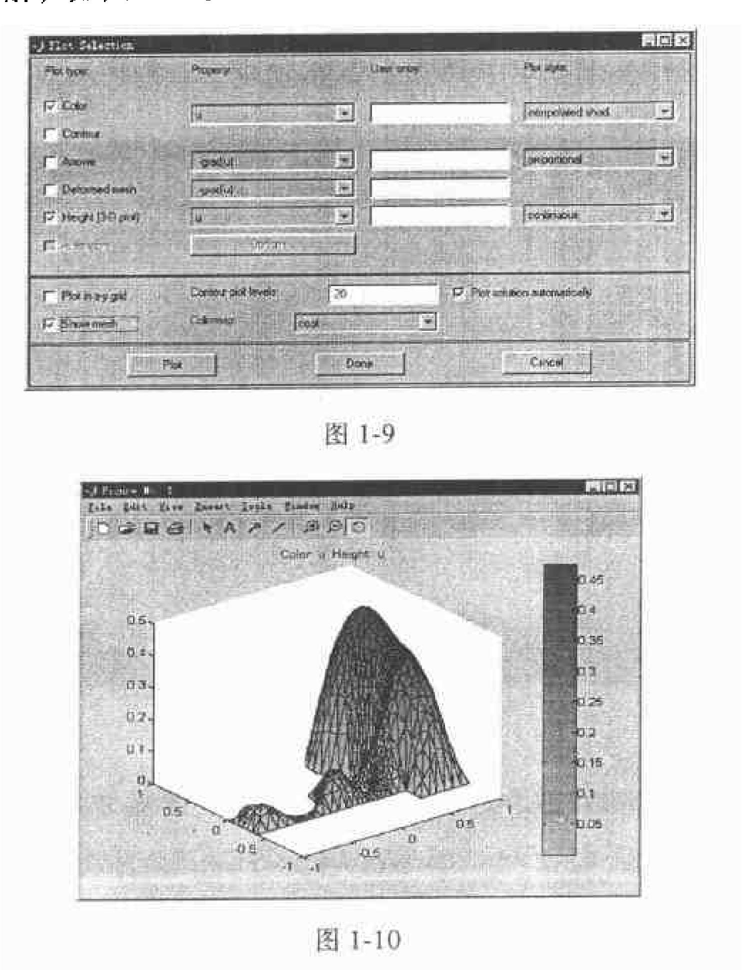


图 1-9

图 1-10

第九步：如果要画等值线图和矢量场图，单击 Plot 菜单中 Parameters...选项，打开 Plot Selection 对话框，选中 Contour 和 Arrows 两项，如图 1-11。然后单击 Plot 按钮，可显示解的等值线图和矢量场图，如图 1-12。

图 1-11

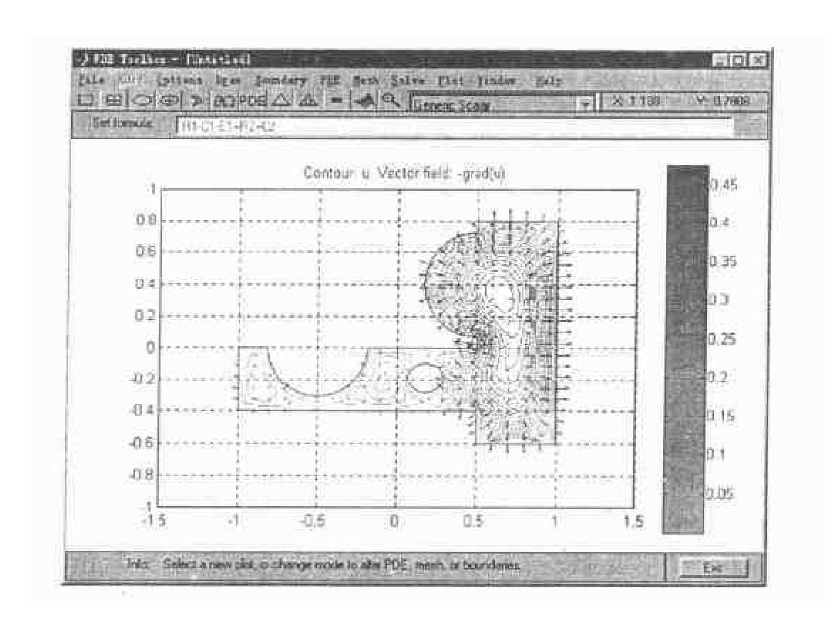


图 1-12

第二章　PDE 图形用户界面

2.1　PDE Toolbox 菜单

1. File菜单（如图2-1）

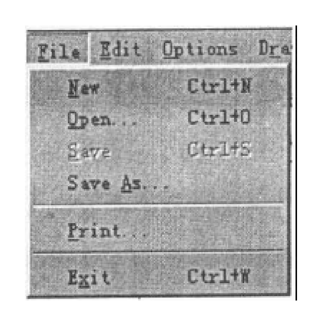


图 2-1

New	新建一个几何结构实体模型(Constructive Solid Geometry，简记为 CSG)、默认文件名为"Untitled"。
Open...	从硬盘装载 M 文件。
Save	将在 GUI 内完成的成果储存到一个 M 文件中。
Save As...	将在 GUI 内完成的成果储存到另外一个 M 文件中。
Print...	将 PDE 工具箱完成的图形送到打印机内进行硬拷贝。
Exit	退出 PDE 工具图形用户界面。

2. Edit 菜单（如图 2-2）

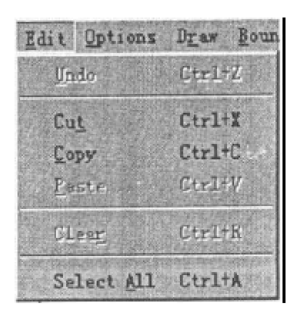


图 2-2

Undo	在绘制多边形时退回到上一步操作。
Cut	将已选实体剪切到剪贴板上。
Copy	将已选实体拷贝到剪贴板上。
Paste...	将剪贴板上的实体粘贴到当前几何结构实体模型中。
Clear	删除已选的实体。
Select All	选择当前几何结构实体造型 CSG 中的所有实体及其边界和子域。

3. Options **菜单**（如图 2-3）

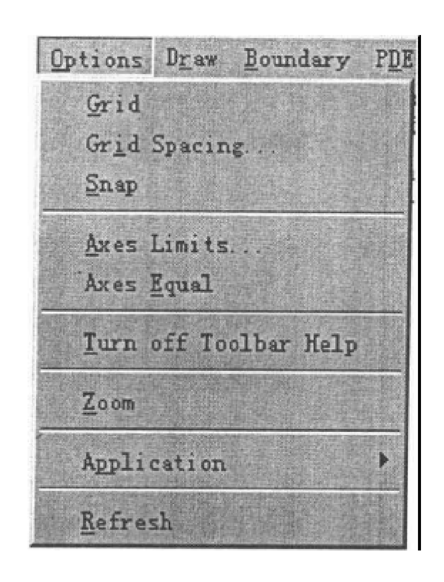


图 2-3

Grid	绘图时打开或关闭栅格。
Grid Spacing...	调整栅格的大小。
Snap	打开或关闭捕捉栅格功能。
Axes Limits...	设置绘图轴的坐标范围。
Axes Equal	打开或关闭绘图方轴。
Turn off Toolbar Help	关闭工具栏按钮的帮助信息。
Zoom	打开或关闭图形缩放功能。
Application	选择应用的模式。
Refresh	重新显示 PDE 工具箱中的图形实体。

4. Draw 菜单（如图 2-4）

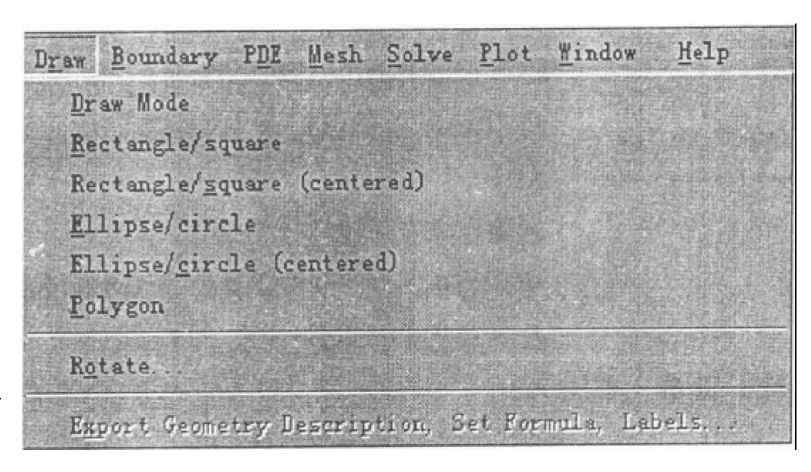


图 2-4

Draw Mode	进入绘图模式。
Rectangle/square	以角点方式画矩形/方形（Ctrl + 鼠标）。
Rectangle/square (centered)	以中心方式画矩形/方形（Ctrl + 鼠标）。
Ellipse/circle	以矩形角点方式画椭圆/圆（Ctrl + 鼠标）。
Ellipse/circle (centered)	以中心方式画椭圆/圆（Ctrl + 鼠标）。
Polygon	画多边形，单击鼠标右键可封闭多边形。
Rotate...	旋转已选的图形。

Export Geometry Description, Set Formula, Labels...

将几何描述矩阵 gd、公式设置字符 sf 和标识空间矩阵 ns 输出到主工作空间去。

单击 Draw 菜单中 Rotate...选项，可打开 Rotate 对话框，通过输入旋转的角度，可使选择的物体按输入的角度逆时针旋转，如图 2-5。旋转中心的选择如果缺省，则为图形的质心，也可以输入旋转中心坐标。

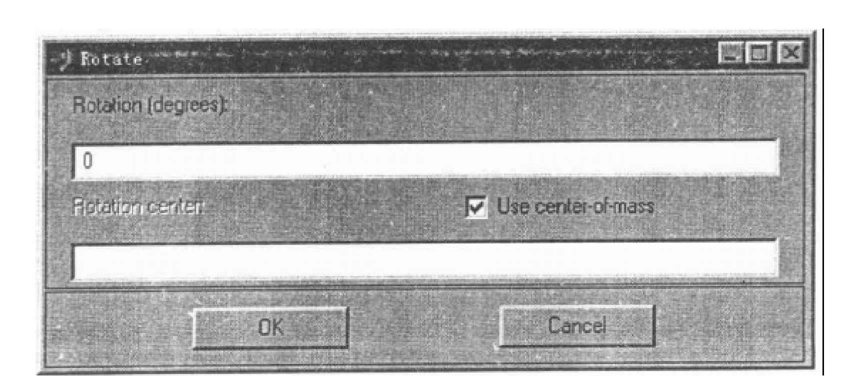


图 2-5

5. Boundary **菜单**（如图 2-6）

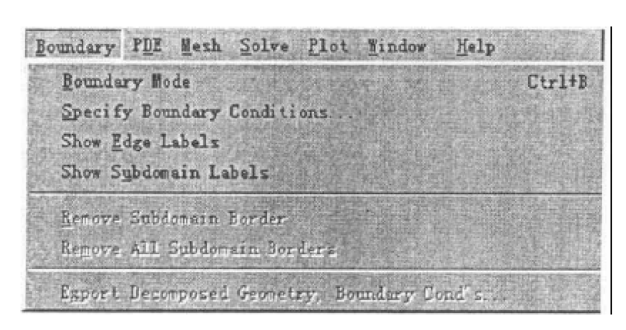


图 2-6

Boundary Mode　进入边界模式。

Specify Boundary Conditions...　对于已选的边界输入条件，如果没有选择边界，则边界条件适用于所有的边界。

Show Edge Labels　显示边界区域标识开关，其数据是分解几何矩阵的列数。

Show Subdomain Labels　显示子区域标识开关，其数据是分解几何矩阵中的子域数值。

Remove Subdomain Border　当图形进行布尔运算时，删除已选取的子域边界。

Remove All Subdomain Borders　当图形进行布尔运算时,删除所有的子域边界。

Export Decomposed Geometry, Boundary Cond's...

将分解几何矩阵 g、边界条件矩阵 b 输出到主工作空间。

选择 Boundary 菜单中 Specify Boundary Conditions...命令可定义边界条件，如图 2-7。在打开的 Boundary Condition 对话框中，可对已选的边界输入边界条件。共有如下三种不同的条件类型：

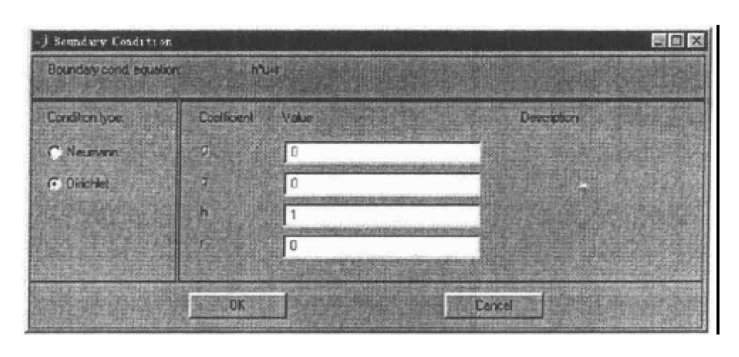


图 2-7

- **一般 Neumann 条件**　这里边界条件是由方程系数 q 和 g 确定的，在方

程组的情况下，q 是 2×2 矩阵，g 是 2×1 矢量。

- **Dirichlet 条件**　u 定义在边界上，边界条件方程是 h*u=r，这里 h 是可以选择的权因子（通常为 1）。在方程组情况下，h 是 2×2 矩阵，r 是 2×1 矢量。
- **混合边界条件**（仅适合于方程组情形）　它是 Dirichlet 和 Neumann 的混合边界条件，q 是 2×2 矩阵，g 是 2×1 矢量，h 是 1×2 矢量，r 是一个标量。

图 2-8 对话框显示了一般偏微分方程组的边界条件。

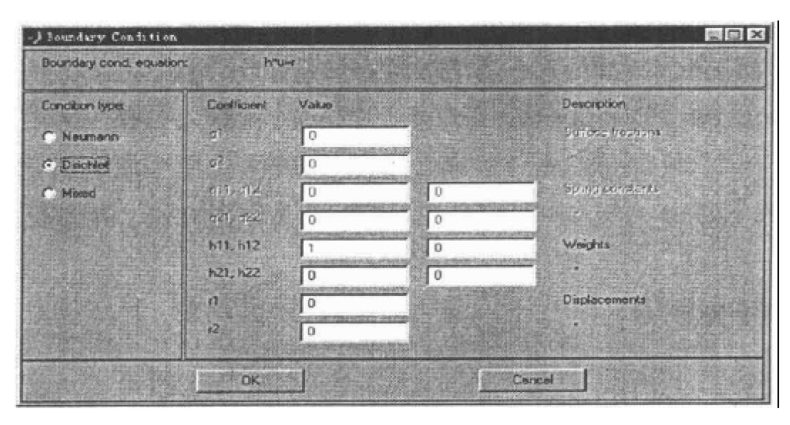


图 2-8

在边界条件输入框中，可以使用如下变量：

- 二维坐标 x 和 y
- 边界线段长度参数 s（s 是以箭头的方向沿边界线段从 0 增加到 1）
- 外法向矢量的分量 nx 和 ny（如果需要边界的切线方向，可以通过 tx=−ny 和 ty=nx 表示）
- 解 u
- 时间 t

注意：如果边界条件是解 u 的函数，必须用非线性求解器。如果边界条件是时间 t 的函数，则必须选择抛物型或者双曲型偏微分方程。

6．PDE **菜单**（如图 2-9）

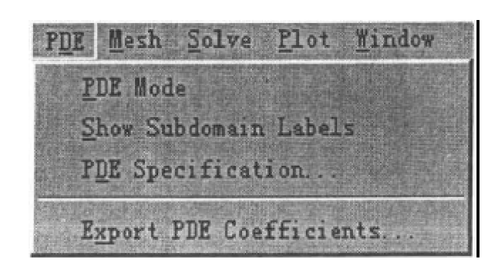


图 2-9

PDE Mode	进入偏微分方程模式。
Show Subdomain Labels	显示子区域标识开关。
PDE Specification...	调整 PDE 参数和类型。
Export PDE Coefficients...	将当前 PDE 参数 c,a,f,d 输出到主工作空间，其参数变量为字符类型。

单击 PDE 菜单中 PDE Specification...选项，可打开如图 2-10 所示的对话框，从中可选择偏微分方程的类型以及对应用参数作一定的调整。参数的维数决定于偏微分方程的维数．如果选择专业应用模型，那么特殊偏微分方程的参数将代替标准偏微分方程系数．每一个参数 c,a,f 或 d 皆可作为有效的 MATLAB 表达式，以及作为计算三角形单元质量中心的参数值。下面的变量是很有用的：x 和 y——点的坐标；u——解；ux,uy——解关于 x 和 y 的导数；t——时间

图 2-10

注意：如果偏微分方程的参数是解 u 或者它的导数 ux 和 uy 的函数，则必须使用非线性求解器；如果偏微分方程参数是时间 t 的函数，则需使用抛物型或双曲型偏微分方程。

也可以输入用户定义的 MATLAB 可接受变量（p, t, u, time）的函数。例如：键入函数 circlef。

c 可以是标量或是 2×2 矩阵。矩阵 c 可以被用于诸如材料各向异性的问题。如果 c 含有两行，则它们是如下 2×2 对角阵的元素 c_{11} 和 c_{22}：

$$\begin{pmatrix} c_{11} & 0 \\ 0 & c_{22} \end{pmatrix};$$

如果 c 为三行，则它们是如下 2×2 对称阵（$c_{21}=c_{12}$）的元素 c_{11}, c_{12} 和 c_{22}：

$$\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}.$$

如果 c 为四行，则它们是 2×2 矩阵的元素 c_{11}, c_{12}, c_{21} 和 c_{22}。

偏微分方程的类型有：

- **椭圆型** 椭圆型偏微分方程的基本类型是

$$-\nabla\cdot(c\nabla u)+au=f,\quad \text{in } \Omega,$$

椭圆型偏微分方程中不含参数 d。

- **抛物型** 抛物型偏微分方程的基本类型是

$$d\frac{\partial u}{\partial t}-\nabla\cdot(c\nabla u)+au=f,\quad \text{in } \Omega,$$

初始条件为 $u_0=u(t_0)$。

- **双曲型** 双曲型偏微分方程的基本类型是

$$d\frac{\partial^2 u}{\partial t^2}-\nabla\cdot(c\nabla u)+au=f,\quad \text{in } \Omega,$$

初始条件为 $u_0=u(t_0)$和 $ut_0=\frac{\partial u}{\partial t}(t_0)$，$x\in\Omega$。

- **特征值问题** 特征值偏微分方程的基本类型是

$$-\nabla\cdot(c\nabla u)+au=\lambda du,$$

特征值偏微分方程中不含参数 f。

对于方程组情况，比如，在 Generic System 应用模型中，c 是一个秩为 4 的张量，可以用 2×2 矩阵[c11,c12;c21,c22]表示；a 和 d 是 2×2 矩阵；f 是 2×1 矢量。偏微分方程组情形的对话框如图 2-11 所示。

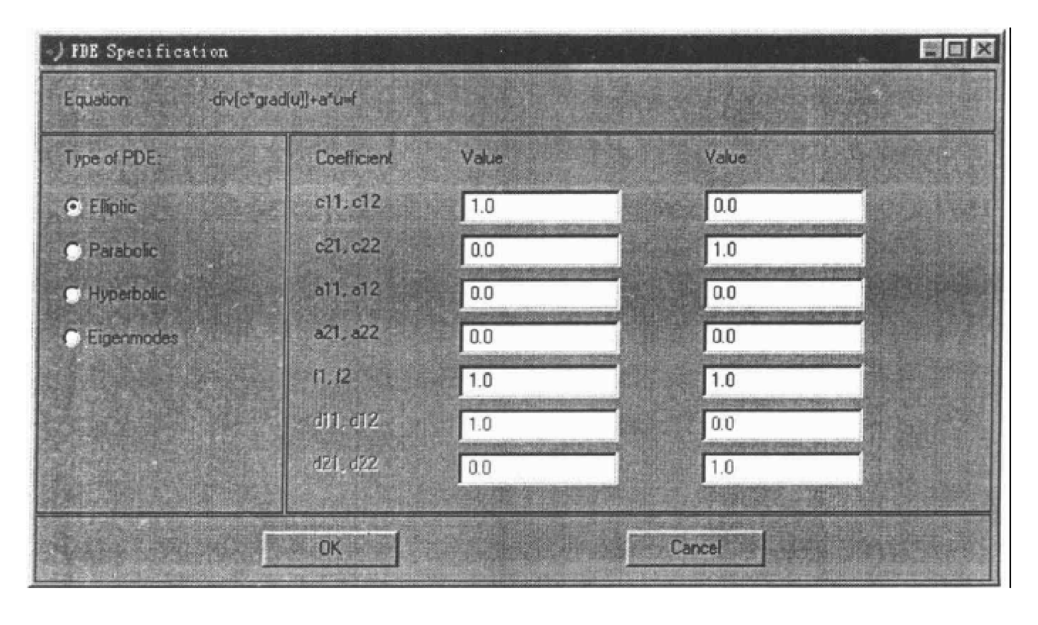


图 2-11

7. Mesh **菜单**（如图 2-12）

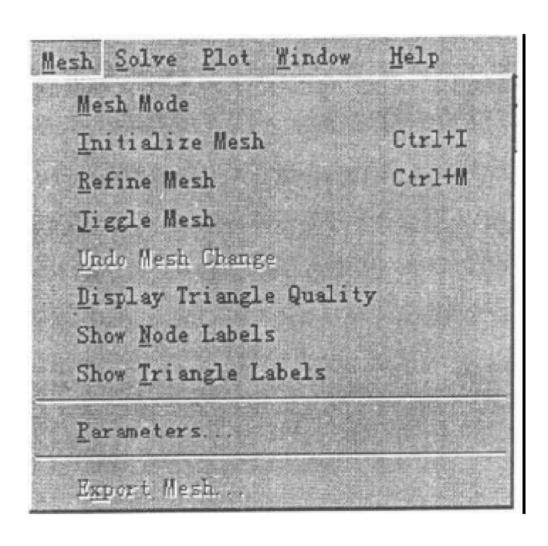


图 2-12

Mesh Mode	输入网格模式。
Initialize Mesh	建立和显示初始化三角形网格。
Refine Mesh	加密当前三角形网格。
Jiggle Mesh	优化网格。
Undo Mesh Change	退回上一次网格操作。
Display Triangle Quality	用 0~1 之间数字化的颜色显示三角形网格的质量，大于 0.6 的网格是可接受的。
Show Node Labels	显示网格节点标识开关，节点标识数据是点矩阵 p 的列。
Show Triangle Labels	显示三角形网格标识开关，三角形网格标识数据是三角形矩阵 t 的列。
Parameters...	修改网格生成参数。
Export Mesh...	输出节点矩阵 p、边界矩阵 e 和三角形矩阵 t 到主工作空间。

单击 Mesh 菜单中 Parameters...选项，打开 Mesh Parameters 对话框，可对网格初始化算法 Initmesh 的参数进行调整，其参数意义如下（如图 2-13）:

- **Maximum edge size**　即三角形最大边的长度，这个参数是可选的，但必须是正的实数。
- **Mesh growth rate**　为网格增长速度。网格的大小随区域的几何尺寸增加而增加。其数值一定在 1～2 之间。缺省数值是 1.3，即网格增长速率为 30%

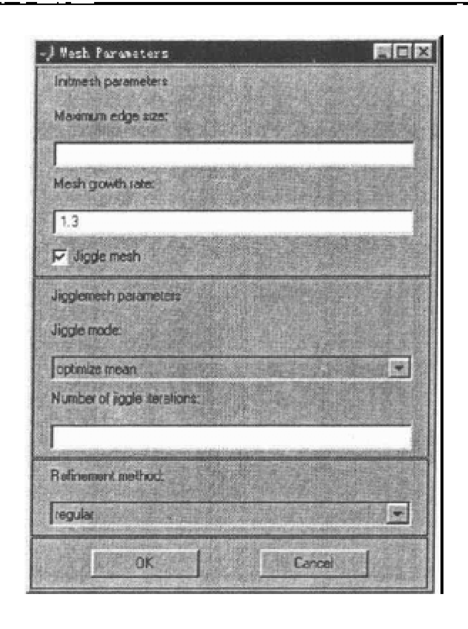


图 2-13

- **Jiggle mesh** 自动优化初始网格开关切换，其算法为 Jigglemesh。

使用优化网格算法Jigglemesh的参数有：

- **Jiggle mode** Jiggle 模式有 on, optimize minimum 和 optimize mean 三种，其意义如下：

on	立刻生成优化网格;
optimize minimum	优化网格反复进行直到达到最小三角形质量或者循环终值;
optimize mean	上述选择同样可适用于 optimize mean，但 optimize mean 只能增加三角形平均质量。

- **Number of jiggle iterations** 设置优化循环次数。对于 optimize minimum 和 optimize mean 方式的循环终值，缺省值为 20。
- **Refinement method** 网格加密算法，可以是 regular 或者 longest。缺省方式是 regular，它可产生均匀网格；而 longest 方式常常以最长边优化每一三角形网格。

8．Solve **菜单**（如图 2-14）

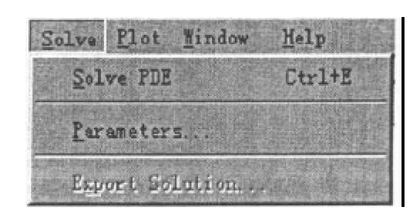


图 2-14

Solve PDE　　对当前的几何结构实体 CSG、三角形网格和图形解偏微分方程。

Parameters...　　调整解 PDE 的参数.

Export Solution...　　输出 PDE 的解矢量 u。如果可行，将计算的特征值 l 输出到主工作空间。

单击 Solve 菜单中 Parameters...选项，打开 Solve Parameters 对话框，从中可输入解方程的参数。每组参数的选择取决于 PDE 的类型。

(1) 对于椭圆型偏微分方程，在缺省方式下，不需要专门定义解方程的参数。解椭圆方程采用基本方程求解器 assempde，在自适应网格生成和 adaptmesh 之间进行选择。对于自适应网格方式，下面的参数可用（如图 2-15）：

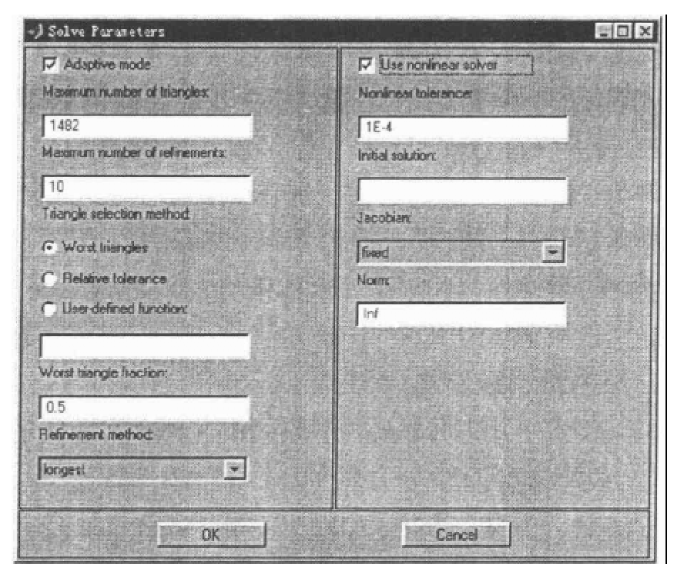


图 2-15

- **Adaptive mode**　打开或关闭自适应方式。
- **Maximum number of triangles**　设置三角形网格允许的最大数目（可以是无穷大），缺省值则是根据当前三角形网格计算的数值。
- **Maximum number of refinements**　设置加密网格最大数目，即试图连续加密网格的最大数目。
- **Triangle selection metbod**　PDE 工具箱中有 3 种三角形网格选择方法：

Worst triangles　　最坏三角形。这个方法是选择比一个最坏三角形的分数值（缺省值为 0.5）还要差的所有三角形。详细情况参见第四章中的 pdeadworst。

Relative tolerance　　相对容差。这个方法使用相对容差标准（缺省值为 1E-4）的三角形，详细情况参见第四章中的

pdeadgsc。

User-defined function 用户自定义函数。输入用户定义的函数和三角形选择方法。参见 pdedemo 的例子。

- **Function parameter** 对函数参数，允许微调三角形选择方法。对于最坏的三角形方法（pdeadworst），它是用于决定三角形需要调整成最坏情形的分数值；对于相对容差方法，它是控制更好地适合于 PDE 的容差参数。

- **Refinement method** 优化方法有两个选项：regular（均匀的）或 longest（最长边的），可参见“Mesh Menu”菜单项中的 Parameters。如果问题是非线性的，即 PDE 参数是直接依赖于解 u 的，那么必须用非线性解，可采用下面的参数：

- **Use nonlinear solver** 选择使用非线性解的切换开关。

- **Nonlinear tolerance** 对于非线性解的容差参数的设置。

- **Initial solution** 一个初始估计值，它可以是当前网格节点上赋以常数 x 和 y 的函数。例如：1 和 exp(x*y)，皆为可选参数，缺省值为零。

- **Jacobian** 为 Jacobian 逼近方法，分 Fixed 固定的（缺省值）、不动点迭代、分块的、分块（对角）逼近或者完全 Jacobian 几种。

- **Nom（范数）** 各种范数是用来计算残差的。取能量范数时输入的值即为能量，或者对于标量 p 给出 l_p 范数，缺省值为 Inf（无穷范数）。

注意：自适应方式和非线性求解器可一起使用。

(2) 对于抛物型偏微分方程（Parabolic PDEs），解抛物型的参数是：

- **Time** 为求解抛物型偏微分方程的 MATLAB 时间矢量。相关时间间隔依赖于问题的动态状况。例如：输入“0:10”或“logspace(-2,0,20)”等。

- **u(t0)** 对于抛物型偏微分方程的初始值是 $u(t_0)$。初始值可以是一个常数或者是当前网格的节点值的列向量。

- **Relative tolerance** 为相对容差。对于常微分方程 ODE 的求解器的相对容差参数，是用来求解抛物型偏微分方程有关时间部分。

- **Absolute tolerance** 为绝对容差。对于常微分方程 ODE 的求解器的绝对容差参数，是用来求解抛物型偏微分方程有关时间部分。

(3) 对于双曲型偏微分方程（Hyperbolic PDEs），解双曲型的参数是：（如图 2-16）

- **Time（时间）** 为求解双曲型偏微分方程的 MATLAB 时间矢量。相关时间间隔依赖于问题的动态状况。比如，Time 项中可设置“0:10”或“logspace(-2,0,20)”。

- **u(t0)** 对于双曲型偏微分方程的初始值是 $u(t_0)$。初始值可以是一个常数或者是当前网格的节点值的列向量。

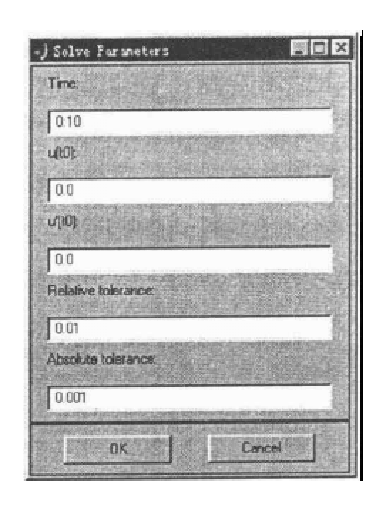


图 2-16

- **u'(t0)**　对于双曲型偏微分方程的初始值是 $\dot{u}(t_0)$，可使用与 $u(t_0)$相同的格式。
- **Relative tolerance** (相对容差)　对于常微分方程 ODE 的求解器的相对容差参数，是用来求解双曲型偏微分方程有关时间部分。
- **Absolute tolerance** (绝对容差)　对于常微分方程 ODE 的求解器的绝对容差参数，是用来求解双曲型偏微分方程有关时间部分。

(4) 对于特征值偏微分方程，解参数仅仅是特征值的求解域，如图 2-17，它是一个二元矢量，在实轴上定义一个区间作为特征值求解域，左端点可以取 -inf (inf 表示无穷 ∞)。比如，特征值的求解域可设置为 "[0 100]" 或 "[-inf 50]" 等。

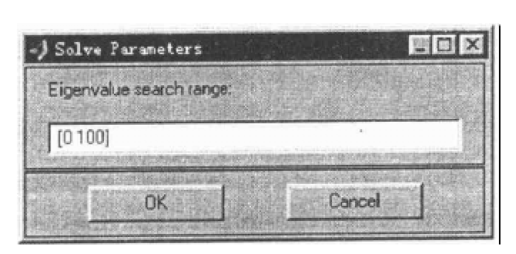


图 2-17

9. Plot **菜单**（如图 2-18）

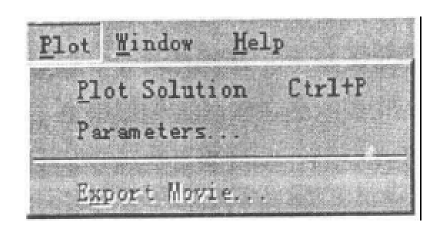


图 2-18

Plot Solution	显示图形解。
Parameters...	打开绘图方式对话框。
Export Movie...	如果动画被录制了，则动画矩阵 M 将输出到主工作空间。

单击 Plot 菜单中 Parameters...选项，可打开 Plot Selection 对话框，如图 2-19，其中含有控制绘图和可视化选项部分。

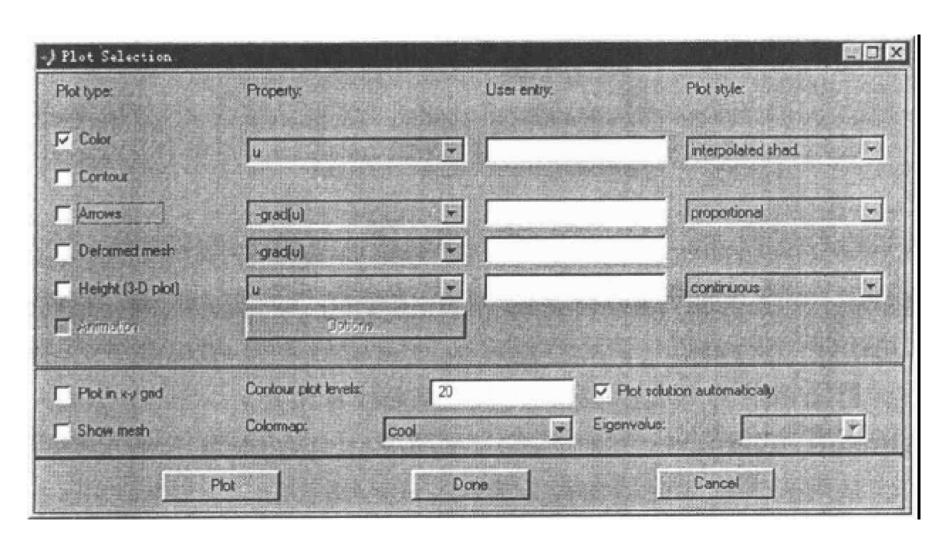


图 2-19 Plot Selection 对话框

- **Plot type**（位于最左列）

有 6 种关于不同绘图方式供选择，可用于可视化。

Color（颜色） 用于着色曲面标量属性的可视化。

Contour（等值线） 用于等值线标量属性的可视化。

当绘图类型（颜色和等值线）被检查后，等值线可提高颜色的可视化，等值线被画成黑色。

Arrows（箭头） 用箭头表示矢量属性的可视化。

Deformed mesh（变形网格） 用向量属性表示变形网格的可视化。变形会自动地控制在问题区域的 10%。这种绘图类型基本上要把结构力学中的 x 和 y 位移（u 和 v）显示出来。如果没有其他的绘图类型选取，那么变形三角形网格会被显示出来。

Height (3-D plot)（三维图形） 分别用不同图形窗口进行三维图形（3-D plot）标量属性的可视化。如果颜色和等值线的绘图类型没有选取，则三维图形（3-D plot）绘出的仅仅是网格图，当然也可以在三维图形（3-D plot）中同时用颜色和(或)等值线绘出其他标量属性。

Animation（动画） 在抛物型和双曲型问题中依赖于时间解的动画。如果

选取了这个选项，方程解就被记录下来，然后用 movie 函数可在不同的图形窗口中做动画演示。

- **Property**（位于第二列）

Property（属性）用于画图时选用相应的绘图类型。

第一个弹出菜单用于控制颜色或等值线的显示属性。其中

u	方程的解。
abs(grad(u))	每个三角形的中心的 ∇u 的绝对值。
abs(c*grad(u))	每个三角形的中心的 $c\cdot\nabla u$ 的绝对值。
user entry	MATLAB 表达式，返回一个定义在当前三角形网格的节点或者三角形上的数据矢量。选取 user entry 项，可以将表达式输入到右边的 User entry 的编辑框中。

u 的导数是 ux 和 uy。$c\cdot\nabla u$ 的分量 cux 和 cuy，以及 x 和 y 都是局部工作空间的变量。

第二和第三个弹出菜单，通过使用箭头和变形网格可视化图形表示矢量值属性。其中

-grad(u)	u 的负梯度，即 $-\nabla u$ 。
-c*grad(u)	c 乘以 u 的负梯度，即 $-c\cdot\nabla u$ 。
user entry	MATLAB 表达式[px; py]可返回一个 $2\times n$ 维定义当前三角形网格数据的矩阵。

解 u，其导数 ux 和 uy，$c\cdot\nabla u$ 的 x 和 y 分量，cux 和 cuy，以及 x 和 y，皆适用于局部空间。三角形中心的值是由节点插值得到的。可以在右边 User entry 中对属性弹出式菜单进行赋值，比如，输入“[ux;uy]”或“[x;y]”。

对于方程组情形，若使用颜色、等值线或三维绘图等可视化属性是 u, v, abs(u,v)和用户输入框。

若使用箭头或变形网格，可选择(u,v)或用户框。对于结构力学中的应用来说，u 和 v 分别是 x 和 y 方向的位移。

- **User entry**（位于第三列）

含有 4 个编辑框，可供用户输入自己的表达式。只有当用户在编辑窗左边的弹出菜单选择了 user entry 项，才可输入属性。

- **Plot style**（位于第四列）

绘图方式含有三个弹出菜单，可分别用作对颜色、箭头及三维绘图的属性控制。

用于绘制彩色（Color）曲面的绘图属性设置是：

interpolated shad.　　插值着色。使用已选的色图 colormap 和插值着

	色，即对于每个三角形区域用线性插值进行着色（缺省值）。
flat shading	使用已选的色图和平坦方式着色，即对每一个三角区域进行单色着色。

对于箭头（Arrows）绘制有两种方式可选择：

proportional	箭头的长度与设置的有关属性大小相对应。
normalized	所有的箭头的长度都是相等的，这对于只想了解矢量场的方向是很有用的，即使区域很小时，矢量的方向也清晰可见。

对于三维绘图 Height (3-D plot)，绘图方式有：

continuous（连续的）	从三角形中点到网格节点用插值方法产生一个光滑的连续曲面。
discontinuous（离散的）	产生一离散曲面，但每一三角区域其数据和高度为常数。

解总共有三个属性（两个标量属性和一个矢量场），可同时显示。如果三维绘图 Height (3-D plot)的选择被关掉，那么在 PDE Toolbox GUI 的主轴上画出二维图形解；否则会在不同的图形窗口内绘出三维图形。

- **辅助绘图控制选择**

在对话框底部有许多辅助绘图控制选择项如下：

Plot in x-y grid 如果选择了此项，图形解将原来的三角形网格转变成矩形 x-y 网格。这对于动画来说是非常有用的，因为四边形网格可有效地加速动画片存储过程。

Show mesh 在曲面图中，如果选择此项，网格被画成黑色；如果缺省，网格消隐。

Contour plot levels 等值线条数，比如，输入 15 或 20，系统会按此数目画出等值线，缺省值为 20。再如，输入“[0:100:1000]”或“logspace(-1,1,30)”等。

Colormap 使用 Colormap 弹出菜单，可以选择不同的色图，如 cool, gray, bone, pink, copper, hot, jet, hsv 和 prism。

Plot solution automatically 如果关闭此项，则 PDE Toolbox 不会立刻显示图解。然而，新解仍然可以用这个对话框画出来。

对于抛物型和双曲型 PDE，对话框中 Animation 选项有效，其右边 Options... 按钮选项中含有动画方面的控制参数（如图 2-20）：

Animation rate (fps)	对于动画参数可控制每秒显示画面数目（fps）。
Number of repeats	设置动画播放循环次数。

Replay movie　　如果选择了此项，当前的动画片将重新播放；如果当前内存没有动画片，这个选择是不可用的。

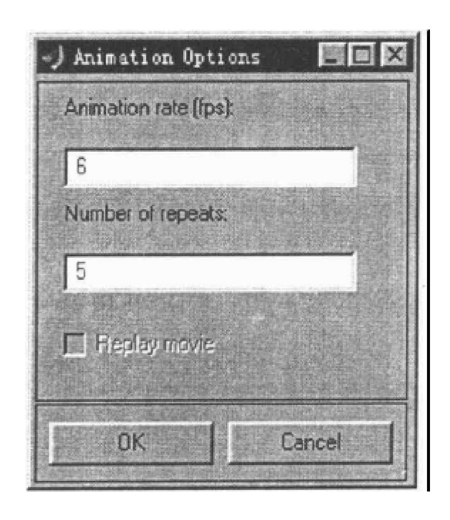


图 2-20

对于特征值问题，右边底部设有关于所有的特征值的弹出式菜单。图形解是关于已选的特征值所对应的特征矢量。缺省时，最小特征值将被选择。

对于三维图形，均处于 Rotated on 状态，用鼠标可动态显示图形。

至此，Plot selection 对话框设置完毕，若点击 Plot 按钮，则按当前图形设置，其解立刻被绘出；如果当前没有偏微分方程，则首先解方程，有了解以后再绘图。

若点击 Done 按钮，对话框则会关闭。当前的设置将被储存，没有新图形产生。

若点击 Cancel 按钮，对话框则会关闭，设置不会改变。

10. Window **菜单**

从 Window 菜单项,可选择当前打开的所有的 MATLAB 图形窗口，被选择的窗口移至前台。

11. Help **菜单**

Help...　　显示帮助信息。

About...　　显示版本信息。

2.2　PDE 工具栏

主菜单下是工具栏，工具栏中含有许多工具图标按钮,可提供快速、便捷的操作方式。从左到右 5 个按钮为绘图模式按钮，紧接着的 6 个为边界、网格、

解方程和图形显示控制功能按钮，最右边的为图形缩放功能键（如图 2-21）：

图 2-21

以角点方式画矩形/方形（Ctrl + 鼠标）。

以中心方式画矩形/方形（Ctrl + 鼠标）。

以矩形角点长轴方式画椭圆/圆（Ctrl + 鼠标）。

以中心方式画椭圆/圆（Ctrl + 鼠标）。

画多边形，按右键可封闭多边形。

进入边界模式。

打开 PDE Specification（偏微分方程类型）对话框。

初始化三角形网格。

加密三角形网格。

解偏微分方程。

打开 Plot Selection 对话框，确定后给出解的三维图形。

为显示缩放切换按钮。

第三章　典型方程及应用实例

求解 PDE 问题主要有两种方法，一种是使用图形用户界面，另一种是采用命令行编程。前者直观简便，而后者更为灵活。

3.1　求解椭圆型方程的例子

【例 3.1-1】　单位圆上的 Poisson 方程边值问题：

$$\begin{cases} -\Delta u = 1, \quad \Omega = \{(x,y) \mid x^2 + y^2 < 1\}, \\ u|_{\partial\Omega} = 0. \end{cases}$$

这一问题的精确解为

$$u(x,y) = \frac{(1-x^2-y^2)}{4}.$$

若使用图形用户界面(Graphical User Interface，简记为 GUI)，则首先在 MATLAB 的工作窗口中键入 pdetool，按回车键确定，于是出现 PDE Toolbox 窗口。如果需要坐标网格，单击 Options 菜单下的 Grid 选项即可。下面分步进行操作。

(i) **画区域圆**　单击工具■，大致在(0,0)位置单击鼠标右键同时拖拉鼠标到适当位置松开，绘制圆。为了保证所绘制的圆是标准的单位圆，在所绘圆上双击，打开 Object Dialog 对话框，精确地输入圆心坐标 X-center 为 0、Y-center 为 0 及半径 Radius 为 1，然后单击 OK 按钮，这样单位圆已画好。

(ii) **设置边界条件**　单击工具■，图形边界变红，逐段双击边界，打开 Boundary Condition 对话框，输入边界条件。对于同一类型的边界，可以按 Shift 键，将多个边界同时选择，统一设置边界条件。本题选择 Dirichlet 条件，输入 h 为 1，r 为 0，然后单击 OK 按钮。也可以单击 Boundary 菜单中 Specify Boundary Conditions...选项，打开 Boundary Condition 对话框输入边界条件，如图 3-1。

(iii) **设置方程**　单击 PDE 菜单中 PDE Specification...选项，打开 PDE Specification 对话框，选择方程类型。本题单击 Elliptic，输入 c 为 1，a 为 0，f 为 1，然后单击 OK 按钮，如图 3-2 所示。

图 3-1

图 3-2

(iv) **网格剖分** 单击工具▦,或者单击 Mesh 菜单中 Initialize Mesh 选项,可进行初始网格剖分,这时在 PDE Toolbox 窗口下方的状态栏内显示出初始网格的节点数和三角形单元数。本题节点数为 144 个,三角形单元数为 254 个。如果需要网格加密,再单击▦,或者单击 Mesh 菜单中 Refine Mesh 选项,这时节点数变为 541 个,三角形单元数为 1 016 个,如图 3-3。如此还可继续加密。

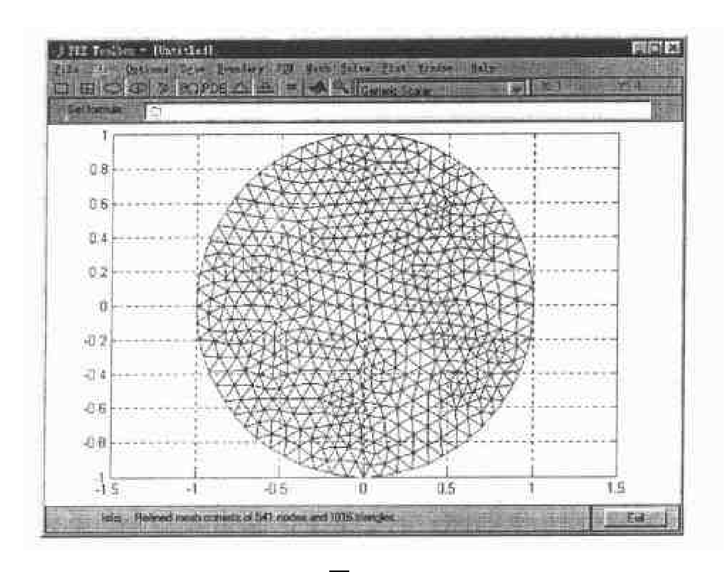


图 3-3

(v) **解方程**　单击工具█，或者单击 Solve 菜单中 Solve PDE 选项，可显示方程色彩解。如果单击 Plot 菜单中 Parameters...选项，出现 Plot Selection 对话框，如图 3-4，从中可以选择 Color，Contour，Arrows，Deformed mesh，Height (3-D plot)，还可以设置等值线的数目等。本例中选择 Color，Contour，Height (3-D plot)和 Show mesh 四项，然后单击 Plot 按钮，方程的图形解如图 3-5 所示．除了作定解问题解 u 的图形外，也可以作|grad u|，|cgrad u|等图形。

图 3-4

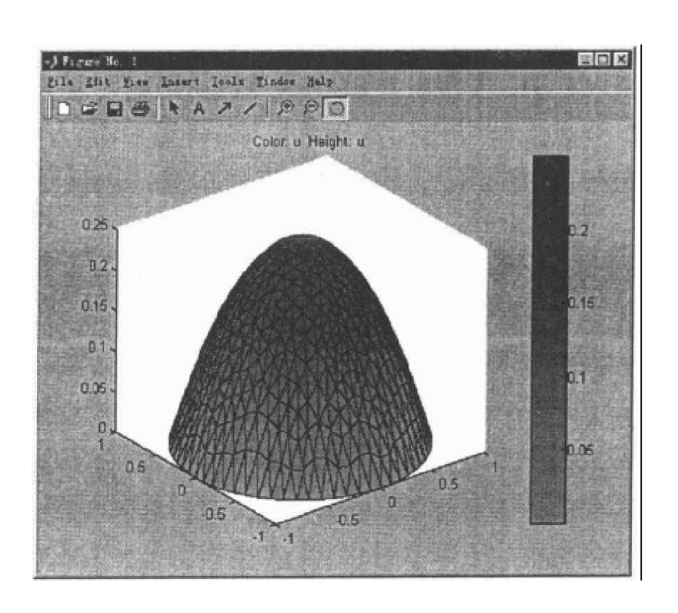


图 3-5

(vi) **与精确解作比较**　单击 Plot 菜单中 Parameters...选项，打开 Plot Selection 对话框，在 Height (3-D plot)行的 Property 下拉框中选 user entry，且在该行的 User entry 输入框中键入 u-(1-x.^2-y.^2)/4，单击 Plot 按钮就可以看到

解的绝对误差图形，如图 3-6。可见在边界处误差为 0。

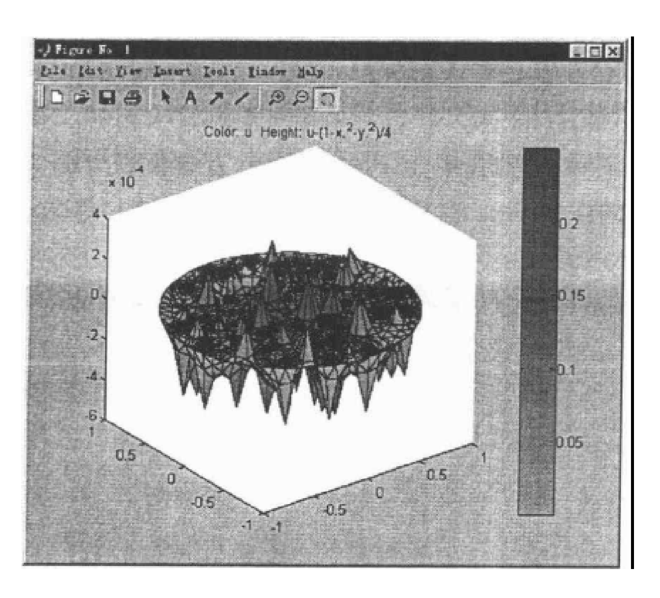


图 3-6

(vii) **输出网格节点的编号、单元编号以及节点坐标** 单击 Mesh 菜单中 Show Node Labels 选项，再单击工具或，即可显示节点编号。若要输出节点坐标，只需单击 Mesh 菜单中 Export Mesh...选项，这时打开的 Export 对话框中的默认值为 p e t，这里 p,e,t 分别表示 points (点)、edges (边)、triangles (三角形)数据的变量，单击 OK 按钮。然后在 MATLAB 命令窗口键入 p，按回车键确定，即可显示出节点按编号排列的坐标(二维数组)；键入 e，按回车键，则显示边界线段数据矩阵(7 维数组)；输入 t，按回车键，则显示三角形单元数据矩阵(4 维数组)。

(viii) **输出解的数值** 单击 Solve 菜单中 Export Solution...选项，在打开的 Export 对话框中输入 u，单击 OK 按钮确定。再在 MATLAB 命令窗口中输入 u，按回车键确定，即显示按节点编号排列的解的数值。

我们也可以用 MATLAB 程序求解 PDE 问题，同时显示解的图形：

```
[p,e,t]=initmesh('circleg','hmax',1);  %初始化网格
error=[];err=1;
while err>0.001,
    [p,e,t]=refinemesh('circleg',p,e,t);  %加密网格
    u=assempde('circleb1',p,e,t,1,0,1);  %求解
    exact=(1-p(1,:).^2-p(2,:).^2)'/4;  %定义精确解
    err=norm(u-exact,'inf');
    error=[error err];
```

```
end
pdemesh(p,e,t)  %绘制网格图
pdesurf(p,t,u)  %绘制解的曲面图
pdesurf(p,t,u-exact)  %绘制误差图形
```

通过命令行键入 help+命令函数，如 help pdemesh，按回车键，可以调入有关命令函数的定义、参数格式等帮助信息。

【例 3.1-2】 散射问题。

设有平面电磁波垂直地射入水平理想金属表面，考虑波速为 c（注意这个常数不能与 PDE Toolbox 中的 c 相混淆）。这时散射波的电矢量也在垂直于水平的方向，则电矢量惟一的分量 $U(x,y,t)$满足波动方程：

$$\frac{\partial^2 U}{\partial t^2}-c^2\Delta U=0.$$

现在考虑单色波$U(x,y,t)=u(x,y)\mathrm{e}^{-k\omega t}$，则 $u(x,y)$满足

$$-\Delta u-\frac{\omega^2}{c^2}u=0.$$

记$\frac{\omega}{c}=k$，于是得到 Helmholtz 方程：

$$-\Delta u-k^2 u=0.$$

这也是椭圆型方程，其中 k 是波数，它与角频率 ω 有关，引入频率 f 和波长 λ，则有

$$k=\frac{\omega}{c}=\frac{2\pi f}{c}=\frac{2\pi}{\lambda}.$$

设扰动的波是平面波

$$V(x,y,t)=\mathrm{e}^{\mathrm{i}(ka\cdot x-\omega t)}=v(x,y)\mathrm{e}^{-\mathrm{i}\omega t},$$

其中 $v(x,y)=\mathrm{e}^{\mathrm{i}ka\cdot x}$。将 u 分成两部分，$u=v+r$，由于入射波 v 满足 Helmholtz 方程，故反射波 r 也满足 Helmholtz 方程。再由于边界上 $u=0$，故 $r=-v(x,y)$，在一定的条件下 r 满足一阶波动方程:

$$\frac{\partial r}{\partial t}+c\boldsymbol{\xi}\cdot\nabla r=0.$$

这里波沿着正的ξ方向传播，ξ为矢径向量。于是得到 r 在边界上满足 Neumann 条件:

$$\boldsymbol{\xi}\cdot\nabla r=\mathrm{i}kr,$$

这里ξ视为区域边界的外法向。

现在我们考虑一个具体的散射问题。一块圆形金属片，中心挖去一正方形，

外边界满足 Neumann 条件，内边界满足 Dirichlet 条件：

$$r = -v(x, y) = -e^{ik\omega \cdot x}.$$

考虑到入射波以 $-x$ 方向，所以上式可写成 $r = -e^{ikx}$，这样得到求解这个反射波 $r(x, y)$ 的定解问题:

$$\begin{cases} \Delta r + k^2 r = 0, & \text{在区域内,} \\ r = -e^{-ikx} = -e^{-i60x}, & \text{内边界上,} \\ \dfrac{\partial r}{\partial n} = -ik = -60i, & \text{外边界上,} \end{cases}$$

这里取 $k = 60$，波长 λ 为 0.1。

现在用 GUI 解这个散射问题。首先在 MATLAB 的工作窗口中键入 pdetool，按回车键确定，调出 PDE Toolbox 窗口，选择 Generic Scalar 模型。如果需要坐标网格，单击 Options 菜单下的 Grid 选项。下面分步进行操作。

(i) **确定定解区域**　先作正方形区域，在 PDE Toolbox 窗口中，单击菜单栏 Draw 下的 Rectangle/square (centered)，或单击工具栏中■，在窗口中用鼠标右键点击正方形中点的大致位置，拖拉出正方形区域，区域为 SQ1。双击 SQ1，在对话框中输入 Left 为 0.75，Bottom 为 0.45，Width 为 0.1，Height 为 0.1，单击 OK 按钮确定。这时出现边长为 0.1、中心坐标为(0.8,0.5)的正方形；当正方形为选中状态，单击 Draw 菜单下的 Rotate...选项，输入 Rotation (degrees)为 45，并选中 Use center-of-mass（以物体中心为旋转中心），然后单击 OK 按钮。这样便形成了旋转后的正方形。

(ii) **作圆域**　单击菜单 Draw 下的 Ellipse/circle（centered）选项，或单击工具栏中■，在窗口中用鼠标右键点击圆心大致位置(0.8,0.5)，拖拉出半径大致为 0.45 的圆 C1。双击 C1，输入圆心坐标 X-center 为 0.8，Y-center 为 0.5，半径 Radius 为 0.45，然后单击 OK 按钮。在工具栏 Set formula 中键入 C1-SQ1，表示圆中挖去正方形，得到定解区域(复连通区域)。

(iii) **设置方程类型与边界条件**　单击菜单 PDE 下的 PDE Specification...选项，在 PDE Specification 对话框中选中 Elliptic，输入 c 为 1，a 为-3600，f 为 0，然后单击 OK 按钮。再确定边界条件：单击工具栏■，使边界变红，逐段双击边界，在对话框中键入边界条件。对外边界，选中 Neumann，输入 q 为-60i，g 为 0；对内边界，选中 Dirichlet，输入 h 为 1，r 为-exp(-i*60*x)。

(iv) **解方程**　单击工具栏■，作网格初始剖分，再单击■，加密剖分。然后单击菜单 Plot 中的 Parameters...选项，在 Plot Selection 对话框中选择 Color，单击 Plot 按钮，立即显示所要求的解的图形，如图 3-7，同时还显示 PDE Toolbox Warning 提示框，表示该解只是实部，虚部没有表示出来。如要作等值线，则

选择 Contour；作矢量线，选择 Arrows；作三维图，选择 Height (3-D plot)。

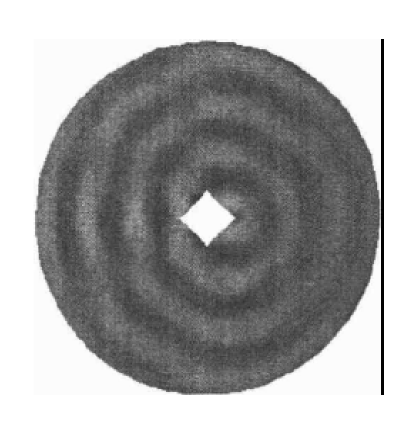


图 3-7

(v) **数值解的输出**　如果要显示出网格剖分数据，在网格剖分状态，单击菜单 Mesh 中 Show Node Labels 选项，可显示节点编号；若单击 Show Triangle Labels 选项，则显示出三角形编号。如果要输出数值解，只要单击 Solve 菜单中 Expert Solution...选项，在 Variable name for solution 栏中输入 u，然后单击 OK 按钮即可。最后在 MATLAB 命令窗口输入 u，按回车键，可立即得到按节点编号的复数解的数值。

其 MATLAB 动画程序参见例 3.8-2。

【例 3.1-3】　最小曲面问题。

在 xy 平面区域 Ω 的边界 $\partial\Omega$ 上给定函数值 $u|_{\partial\Omega}=\varphi(x,y)$，即给定空间一条曲线：

$$\{(x,y,u):\ u=\varphi(x,y),\ (x,y)\in\partial\Omega\}.$$

最小曲面问题就是求张在这一曲线上的曲面，使曲面的面积最小：

$$J(u)=\inf\iint_{\Omega}\sqrt{1+\left(\frac{\partial v}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2}\mathrm{d}x\mathrm{d}y,$$

$$\forall v\in\left\{v\in C^1(\overline{\Omega}),\ \ v|_{\partial\Omega}=\varphi(x,y)\right\}.$$

这一问题可化成一个非线性边值问题。假定

$$\Omega=\{(x,y)\,|\,x^2+y^2<1\},\quad u|_{\partial\Omega}=x^2.$$

这时最小曲面问题等价于一个非线性边值问题:

$$\begin{cases}-\nabla\cdot\left(\dfrac{1}{\sqrt{1+|\nabla u|^2}}\nabla u\right)=0,\ \text{in}\ \Omega,\\ u|_{\partial\Omega}=x^2.\end{cases}$$

下面使用 GUI 来求解这一问题。首先在 PDE Toolbox 窗口的工具栏中选择 Generic Scalar 模式，然后作单位圆(参见例 3.1-1)。

(i) **设置边界条件**　单击，边界变红，双击各段边界，打开 Boundary Condition 对话框，选择 Dirichlet 条件，输入 h 为 1，r 为 x.^2，然后单击 OK 按钮。

(ii) **设置方程类型**　单击，在 PDE Specification 对话框中选择方程类型为 Elliptic，输入 c 为 1./sqrt(1+ux.^2+uy.^2)，a 为 0，f 为 0。

(iii) **网格剖分**　单击。若需要加密剖分，单击。

(iv) **解方程**　单击 Solve 菜单中 Parameters...选项，打开 Solve Parameters 对话框，选 Use nonlinear solver (使用非线性求解器)项，Nonlinear tolerance (非线性解的容差)设置为 1E-3（即 10^{-3}）。下面解方程。单击，出现解的彩色图形。或者单击 Plot 菜单下的 Parameter...选项，或单击，打开 Plot Selection 对话框，根据要求选择 Color 等项，这里我们选择 Color、Contour、Arrows、Height (3-D plot)及 Show mesh 五项，再单击，可得到带有平面等值线及网格线的三维图形，如图 3-8。

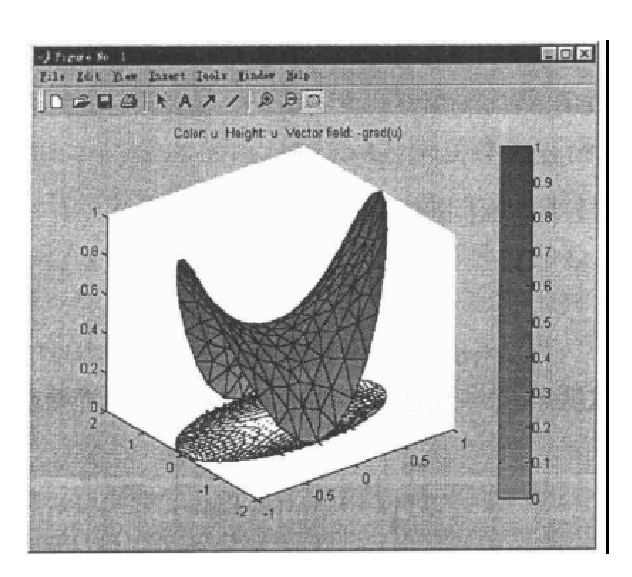


图 3-8

其 MATLAB 程序如下：

```
g='circleg';   %将定义区域文件circleg赋值给符号变量g
b='circleb2';   %定义符号变量b为边界文件
c='1./sqrt(1+ux.^2+uy.^2)';
rtol=1E-3;   %定义相对误差精度
[p,e,t]=initmesh(g);   %在g上网格初始化
[p,e,t]=refinemesh(g,p,e,t);   %加密网格
```

```
u=pdenonlin(b,p,e,t,c,0,0,'Tol',rtol);  %用非线性求解器
    %pdenonlin求解
pdesurf(p,t,u)  %绘制解u的图形
```

【例 3.1-4】　非线性椭圆型方程的例子。

考虑非线性椭圆型方程

$$-\nabla\cdot(c(u)\nabla u)+a(u)u-f(u)=0$$

的定解问题。

用 GUI 求解，单击菜单 Solve 下的 Parameters...选项，选 Use nonlinear solver 项，然后单击 OK 按钮，就可以解非线性问题了。比如，在单位圆上求解

$$-\nabla\cdot(\nabla u)=-16\sqrt{u+1},\quad \text{in}\quad \Omega,$$

$$u=0,\quad \text{on}\quad \partial\Omega.$$

这一问题的精确解为 $u=(x^2+y^2)^2-1$。在设置 Use nonlinear solver 时，其选项 Nonlinear tolerance 设置为 1E-1。再作单位圆区域，将方程类型设置为 Elliptic，参数 c=1，a=0，f=-16*(u+1).^0.5。对边界条件进行设置：单击$\partial\Omega$，再单击菜单 Boundary 下的 Specify Boundary Conditions...选项，打开 Boundary Condition 对话框，选 Dirichlet 条件，输入 h 为 1，r 为 0。再作网格剖分。最后求解：单击 Plot 菜单中 Parameters...选项，打开 Plot Selection 对话框，选 Color, Contour, Height (3-D plot)，单击 Plot 按钮，即显示出解 u 的三维彩色图形，如图 3-9。

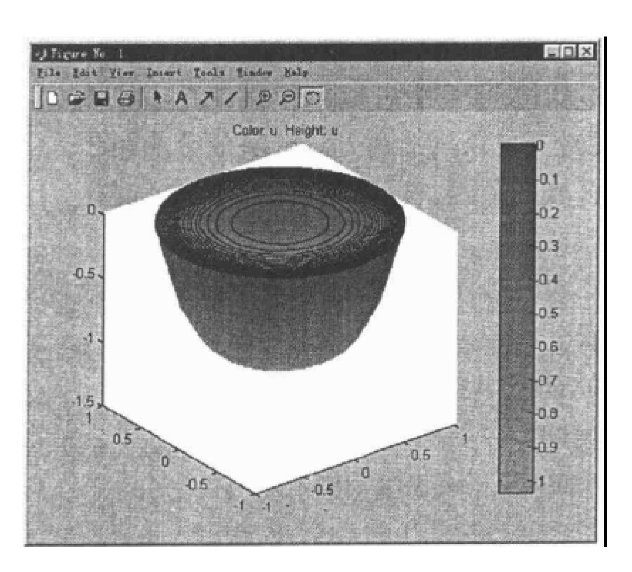


图 3-9

注意：本题中如果将 Nonlinear tolerance 设置为 1E-2 或者更高精度，则会显示 PDE Toolbox Error 提示框，显示 Too many iterations，而无法计算出结果。本题精确解已知，如果在 Plot Selection 对话框中 Color 栏对应的 Property 项中

选 user entry，并在其右侧的 User entry 栏中键入计算值的误差：u-(x.^2+y.^2).^2+1，按 Plot 按钮确定，则显示出用高度表示的解 u 以及用颜色表示的误差，可以看出误差不超过 0.03。

【例 3.1-5】 区域分解方法。

在 PDE Toolbox 中设计了单层区域分解方法。如果 Ω 是一个复杂的几何区域，它可以分解成若干个比较简单的子区域的并集。

假定 Ω 是某些不相交子区域 $\Omega_1,\Omega_2,\cdots,\Omega_n$ 的并集，那么可以对 Ω 上的网格节点重新编号，使得每个子区域的节点的标号统一起来，而且对于两个或两个以上子区域的共同节点的所有标号排在最后。由于 K 在相邻节点编号形成的同一行或同一列不能全为 0，则刚度矩阵可以分裂成如下形式：

$$K=\begin{pmatrix} K_1 & 0 & \dots & B_1^{\mathrm{T}} \\ 0 & K_2 & \dots & B_2^{\mathrm{T}} \\ \vdots & \vdots & & \vdots \\ B_1 & B_2 & \dots & C \end{pmatrix}.$$

负荷向量为

$$F=\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \\ f_c \end{pmatrix}.$$

PDE Toolbox 的程序 assempde 能自动组装 K_j,B_j,f_j,C。最后，线性方程组 $KU=F$ 中的刚度矩阵可以按上述方法分解成分块矩阵。

现在考虑 L 形区域的薄膜，可以利用如下命令作图：

```
pdegplot('lshapeg')
```

注意到子区域之间的边界。现在有 3 个子区域，故上述公式中 $n=3$。对这个几何区域生成网格：

```
[p,e,t]=initmesh('lshapeg');
[p,e,t]=refinemesh('lshapeg',p,e,t);
[p,e,t]=refinemesh('lshapeg',p,e,t);
```

现在对 $n=3$，有

$$\begin{pmatrix} K_1 & 0 & 0 & B_1^{\mathrm{T}} \\ 0 & K_2 & 0 & B_2^{\mathrm{T}} \\ 0 & 0 & K_3 & B_3^{\mathrm{T}} \\ B_1 & B_2 & B_3 & C \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_c \end{pmatrix}=\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_c \end{pmatrix}.$$

按块消元求解得到

$$(C - B_1K_1^{-1}B_1^{\mathrm{T}} - B_2K_2^{-1}B_2^{\mathrm{T}} - B_3K_3^{-1}B_3^{\mathrm{T}})u_c = f_c - B_1K_1^{-1}f_1 - B_2K_2^{-1}f_2 - B_3K_3^{-1}f_3,$$

$$u_1 = K_1^{-1}(f_1 - B_1^{\mathrm{T}}u_c),$$

$$\cdots.$$

在 MATLAB 中可采用比较有效的 Choleski 分解算法：

```
[p,e,t]=initmesh('lshapeg');
time=[];
np=size(p,2);
%求公共点
c=pdesdp(p,e,t);

nc=length(c);
C=zeros(nc,nc);
FC=zeros(nc,1);

[i1,c1]=pdesdp(p,e,t,1);ic1=pdesubix(c,c1);
[K,F]=assempde('lshapeb',p,e,t,1,0,1,time,1);
K1=K(i1,i1);d=symmmd(K1);i1=i1(d);
K1=chol(K1(d,d));B1=K(c1,i1);a1=B1/K1;
C(ic1,ic1)=C(ic1,ic1)+K(c1,c1)-a1*a1';
f1=F(i1);e1=K1'\f1;FC(ic1)=FC(ic1)+F(c1)-a1*e1;

[i2,c2]=pdesdp(p,e,t,2);ic2=pdesubix(c,c2);
[K,F]=assempde('lshapeb',p,e,t,1,0,1,time,2);
K2=K(i2,i2);d=symmmd(K2);i2=i2(d);
K2=chol(K2(d,d));B2=K(c2,i2);a2=B2/K2;
C(ic2,ic2)=C(ic2,ic2)+K(c2,c2)-a2*a2';
f2=F(i2);e2=K2'\f2;FC(ic2)=FC(ic2)+F(c2)-a2*e2;

[i3,c3]=pdesdp(p,e,t,3);ic3=pdesubix(c,c3);
[K,F]=assempde('lshapeb',p,e,t,1,0,1,time,3);
K3=K(i3,i3);d=symmmd(K3);i3=i3(d);
K3=chol(K3(d,d));B3=K(c3,i3);a3=B3/K3;
C(ic3,ic3)=C(ic3,ic3)+K(c3,c3)-a3*a3';
f3=F(i3);e3=K3'\f3;FC(ic3)=FC(ic3)+F(c3)-a3*e3;
```

```
%求解
u=zeros(np,1);
u(c)=C\FC;
u(i1)=K1\(e1-a1'*u(c1));
u(i2)=K2\(e2-a2'*u(c2));
u(i3)=K3\(e3-a3'*u(c3));
%这一问题也可以不用子区域分解方法求解
%与不使用子区域的方法求解的比较
[K,F]=assempde('lshapeb',p,e,t,1,0,1);u1=K\F;
norm(u-u1,'inf')
pdesurf(p,t,u)
```

3.2 求解抛物型方程的例子

【例 3.2-1】 热传导方程：金属板的导热问题。

考虑一个带有矩形孔的金属板上的热传导问题。板的左边保持在 100℃，板的右边热量从板向环境空气定常流动，其他边及内孔边界保持绝缘。初始 $t=t_0$ 时板的温度为 0℃，于是概括为如下定解问题：

$$\begin{cases} d\dfrac{\partial u}{\partial t}-\Delta u=0, & \\ u=100, & \text{左边界上，} \\ \dfrac{\partial u}{\partial n}=-1, & \text{右边界上，} \\ \dfrac{\partial u}{\partial n}=0, & \text{其他边界上，} \\ u\big|_{t=t_0}=0. & \end{cases}$$

域 Ω 的外边界顶点坐标为(−0.5,−0.8)，(0.5,−0.8)，(0.5,0.8)，(−0.5,0.8)。内边界顶点坐标为(−0.05,−0.4)，(0.05,−0.4)，(0.05,0.4)，(−0.05,0.4)。

使用 GUI 求解这一问题。在 PDE Toolbox 窗口的工具栏中选择 Generic Scalar 模式。

(i) **区域设置** 单击▇工具，在窗口拖拉出一个矩形，双击矩形区域，在 Object Dialog 对话框中输入 Left 为-0.5, Bottom 为-0.8, Width 为 1, Height 为 1.6，单击 OK 按钮，显示矩形区域 R1。用同样方法作内孔 R2，只要设置 Left=-0.05, Bottom=-0.4, Width=0.1, Height=0.8 即可。然后在 Set formula 栏中键入 R1-R2。

(ii) **设置边界条件** 单击▇，使边界变红色，然后分别双击每段边界，打开 Boundary Condition 对话框，设置边界条件。在左边界上，选择 Dirichlet

条件，输入 h 为 1, r 为 100；右边界上，选择 Neumann 条件，输入 g 为-1, q 为 0；其他边界上，选择 Neumann 条件，输入 g 为 0, q 为 0。

(iii) **设置方程类型**　单击，打开 PDE Specification 对话框，设置方程类型为 Parabolic (抛物型), d=1, c=1, a=0, f=0，单击 OK 按钮。

(iv) **网格剖分**　单击，或者加密网格，单击。

(v) **初值和误差的设置**　单击 Solve 菜单中 Parameters...选项，打开 Solve Parameters 对话框，输入 Time 为 0:5, u(t0)为 0, Relative tolerance 为 0.01, Absolute tolerance 为 0.001，然后单击 OK 按钮。

(vi) **数值解的输出**　单击 Solve 菜单中 Export Solution...选项，在 Export 对话框中输入 u，单击 OK 按钮。再在 MATLAB 命令窗口中键入 u，按回车键，这时显示出按节点编号的数值解。

(vii) **解的图形**　单击 Plot 菜单中 Parameters...选项，打开 Plot Selection 对话框，选 Color, Contour, Arrows，单击 Plot 按钮，窗口显示出 Time=5 时解的彩色图形，如图 3-10。

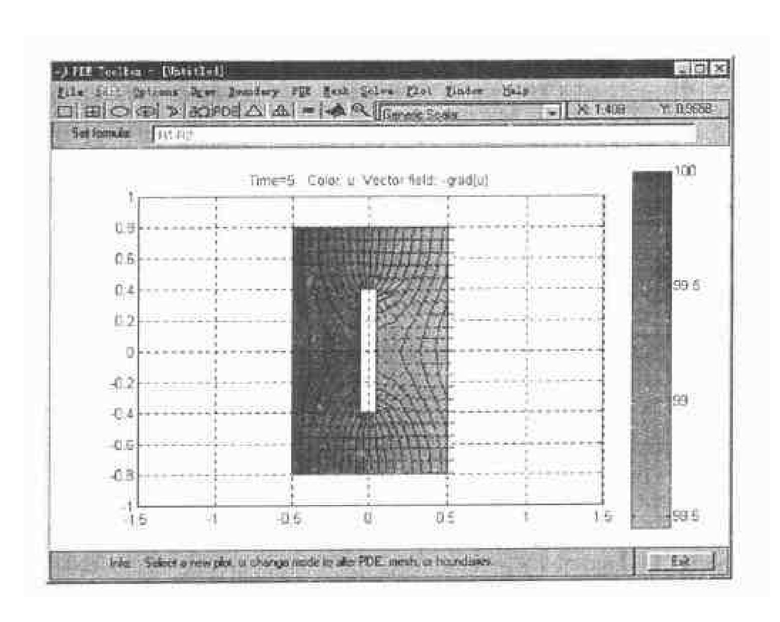


图 3-10

MATLAB 程序：

```
[p,e,t]=initmesh('crackg');  %网格初始化
u=parabolic(0,0:0.5:5,'crackb',p,e,t,1,0,0,1);  %求解
pdeplot(p,e,t,'xydata',u(:,11),'mesh','off',...
    'colormap','hot')  %绘图
```

【例 3.2-2】　放射性杆的热扩散。

这是一个三维的抛物型方程问题，现在利用柱坐标将它简化成二维问题。考虑一个圆柱形放射性杆，其左端供热，右端保持常温，侧面与环境有热交换。由于放射性作用，热量均匀地产生。初始温度为 0℃。于是可以用如下方程描述：

$$\rho c\frac{\partial u}{\partial t}-\nabla\cdot(k\nabla u)=f\ ,$$

其中 ρ 为密度，c 为杆的热容量，k 为导热系数，f 为放射性热源密度。这一金属杆的密度 ρ 取为 7 800 kg/m^3，热容量 c 为 500 ws/kg℃，导热系数 k 为 40 w/m℃，热源密度 f 为 20 000 w/m^3，右端恒温为 100℃，侧面环境温度为 100℃，热交换系数为 50 w/m^2℃，左端的热流为 5 000 w/m^2。

现在取柱坐标 (r,θ,z)，由于关于轴的对称性，故与 θ 无关，从而化为仅与 (r,z) 有关的二维方程：

$$r\rho c\frac{\partial u}{\partial t}-\frac{\partial}{\partial r}\left(kr\frac{\partial u}{\partial r}\right)-\frac{\partial}{\partial z}\left(kr\frac{\partial u}{\partial z}\right)=fr\,.$$

边界条件(如图 3-11)：

在杆的左端($z=-1.5$ 处)：

$$\boldsymbol{n}\cdot(k\nabla u)=5000\ ,$$

由于 PDE Toolbox 中方程形式为

$$\boldsymbol{n}\cdot(c\nabla u)+qu=g\ ,$$

c 与 r 有关：$c=kr$，因此边界条件可写成

$$\boldsymbol{n}\cdot(c\nabla u)=5000r\,.$$

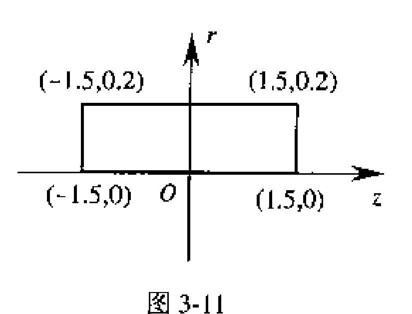


图 3-11

在杆的右端($z=1.5$ 处)：$u=100$。

在杆的侧面($r=0.2$ 处)：$\boldsymbol{n}\cdot(k\nabla u)=50(100-u)$，即

$$\boldsymbol{n}\cdot(c\nabla u)+50r\cdot u=50r\cdot 100\,.$$

在杆的轴心($r=0$ 处)：属于非边界，需要给出人为强制边界条件

$$\boldsymbol{n}\cdot(c\nabla u)=0\,.$$

初始温度：$u|_{t=t_0}=0$。

下面利用 GUI 解这一问题。将杆的轴心 z 方向记为 x 轴，杆的半径 r 方向记为 y 轴。

(i) **区域设置**　按例 3.2-1 的方法作矩形，设置端点坐标为(−1.5,0), (1.5,0), (1.5,0.2), (−1.5,0.2)。在 Object Dialog 对话框中输入 Left 为-1.5, Bottom 为 0, Width 为 3, Height 为 0.2，单击 OK 按钮。形成长为 3、半径为 0.2 的杆的半截面区域，如图 3-11。

(ii) **设置边界条件**　单击▓，使边界变红。双击边界，打开 Boundary Condition 对话框。左边界用 Neumann 条件，输入 q 为 0, g 为 5000*y；右边界

用 Dirichlet 条件，输入 h 为 1, r 为 100；侧面边界用 Neumann 条件，输入 q 为 50*y, g 为 50*y*100；轴心用 Neumann 条件，输入 q 为 0, g 为 0。

(iii) **设置方程类型**　单击，打开 PDE Specification 对话框，设置方程为 Parabolic (抛物型), c=40*y, a=0, d=7800*500*y, f=20000*y。

(iv) **解的参数设置**　单击 Solve 菜单中 Parameters...选项，打开 Solve Parameters 对话框，设置 Time=0:20000，u(t0)=0, Relative tolerance=0.01, Absolute tolerance= 0.001，单击 OK 按钮。

(v) **网格剖分**　单击，若需加密网格，单击。

(vi) **解的图形**　单击 Plot 菜单中 Parameters...选项，打开 Plot Selection 对话框，选 Color, Contour，单击 Plot 按钮，窗口显示出 Time=20000 时解的彩色图形，如图 3-12。

还可以设置动画：按每 1000 秒显示一次，计算到 20000 秒。我们可以看到热量的流动。为此需单击 Plot 菜单中 Parameters...选项，在打开的 Plot Selection 对话框中，选 Animation。

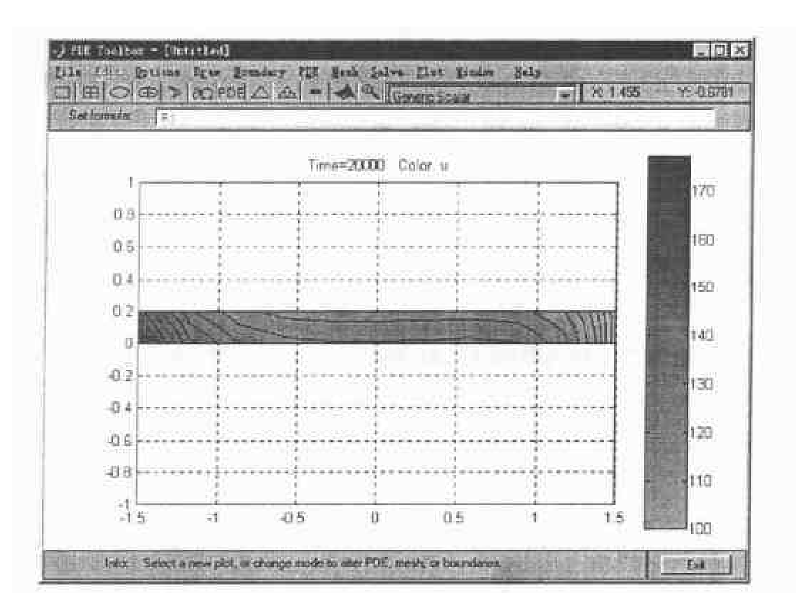


图 3-12

如果在 PDE Specification 对话框中设置方程类型为 Elliptic，这时可得到定常问题的解，也就是抛物型方程在 $t \to +\infty$ 的解。

3.3　求解双曲型方程的例子

考虑如下二维波动方程的定解问题：

$$
\begin{cases}
\dfrac{\partial^2 u}{\partial t^2}-\Delta u=0, & (x,y)\in \Omega=\{(x,y)\mid -1<x<1,\ -1<y<1\},\\
u\mid_{x=\pm 1}=0, & \\
\left.\dfrac{\partial u}{\partial n}\right|_{y=\pm 1}=0, & \\
t=0, & u=\arctan\left(\cos\left(\dfrac{\pi}{2}x\right)\right),\\
 & \dfrac{\partial u}{\partial t}=3\sin(\pi x)\,e^{\sin\left(\frac{\pi}{2}y\right)}.
\end{cases}
$$

用 GUI 求解。类似前面的例子，首先作正方形区域：设置端点坐标为(−1,1), (−1, −1), (1, −1), (1,1)。在 Object Dialog 对话框中输入 Left 为-1, Bottom 为-1, Width 为 2, Height 为 2，单击 OK 按钮。设置边界条件：左、右边界用 Dirichlet 条件，输入 h 为 1, r 为 0；上、下边界用 Neumann 条件，输入 q 为 0, g 为 0。作网格剖分。设置方程类型为 Hyperbolic (双曲型)，键入 c=1, a=0, f=0, d=1。

单击 Solve 菜单中 Parameters...选项，打开 Solve Parameters 对话框，在 Time 栏中键入 linspace(0,5,31)，设置 u 的初始值 u(t0)为 atan(cos(pi/2*x))，u'的初始值 u'(t0)为 3*sin(pi*x).*exp(sin(pi/2*y))，Relative tolerance 为 0.01，Absolute tolerance 为 0.001。

最后输出图形解：单击 Plot 菜单中 Parameters...选项，打开 Plot Selection 对话框，选 Color, Contour, Arrows, Height (3-D plot)和 Show mesh 五项，单击 Plot 按钮，三维彩色图形的解如图 3-13 所示。

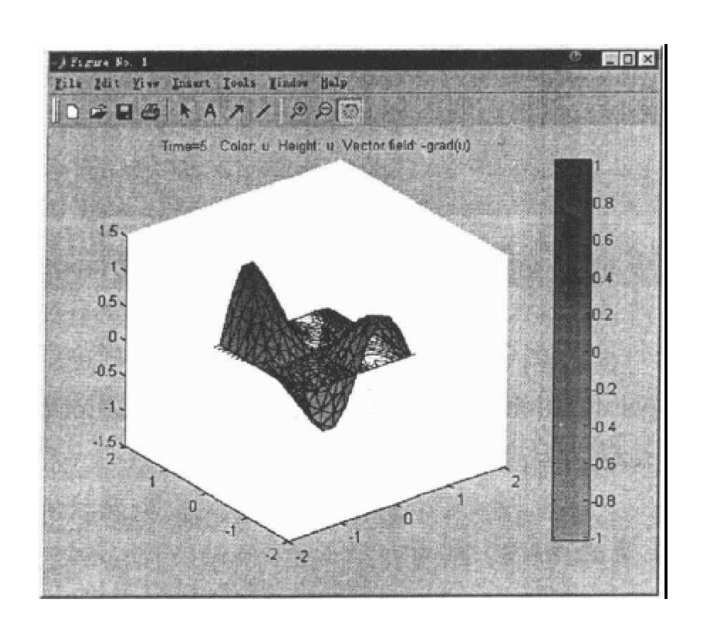


图 3-13

MATLAB的动画程序:

```
[p,e,t]=initmesh('squareg');  %初始化网格
x=p(1,:)';y=p(2,:)';
u0=atan(cos(pi/2*x));  %定义初始条件
ut0=3*sin(pi*x).*exp(sin(pi/2*y));  %定义初始条件
n=31;
tlist=linspace(0,5,n);  %列举时间表格
uu=hyperbolic(u0,ut0,tlist,'squareb3',p,e,t,1,0,0,1);  %求解
delta=-1:0.1:1;
[uxy,tn,a2,a3]=tri2grid(p,t,uu(:,1),delta,delta);  %将三角网
   %格转化为矩形网
gp=[tn;a2;a3];
umax=max(max(uu));
umin=min(min(uu));

newplot
M=moviein(n);
for i=1:n,
    pdeplot(p,e,t,'xydata',uu(:,i),'zdata',uu(:,i),...
    'mesh','off','xygrid','on','gridparam',gp,...
    'colorbar','off','zstyle','continuous');
    axis([-1 1 -1 1 umin umax]); caxis([umin umax]);
    M(:,i)=getframe;
end
movie(M,10);
```

3.4　求解特征值问题的例子

先考虑一个方域上的特征值问题:

$$\begin{cases}-\Delta u=\lambda u, & (x,y)\in\Omega=\{(x,y)\mid 0<x,y<1\},\\ u|_{\partial\Omega}=0.\end{cases}$$

这个问题有精确解：特征值为 $\lambda_{m,n}=\pi^2(m^2+n^2)$ ， $m,n=1,2,3,\cdots$ ，对应的特征函数为 $u_{m,n}=\sin\pi mx\sin\pi ny$。

现在用 GUI 求解。在 PDE Toolbox 窗口使用 Generic Scalar 模式。

先作方域：4 个端点坐标为(0,0), (1,0), (1,1), (0,1)，在 Object Dialog 对话框中输入 Left 为 0, Bottom 为 0, Width 为 1, Height 为 1。设置边界条件为 Dirichlet 条件，输入 h 为 1, r 为 0。下面定义特征值问题：单击■, 在 PDE Specification

对话框中选 Eigenmodes (特征值模式)，PDE 的系数设置为 c=1, a=0, d=1，单击 OK 按钮。再设置特征值范围，如考虑求小于 100 的特征值，单击 Solve 菜单中 Parameters...选项，在打开的 Solve Parameters 对话框中键入[0 100]，单击 OK 按钮确定。

作网格剖分：单击，再加密剖分，单击。

在 Plot Selection 中设置对图形的显示要求：选 Contour 一项，单击 Done 按钮确定。

为在窗口中最大范围地显示图形，单击 Options 菜单下 Axes Limits...选项，打开 Axes Limits 对话框，设置 X-axis range (X 轴范围)为[0 1]，Y-axis range (Y 轴范围)为[0 1]。再单击 Options 菜单下 Axes Equal 选项，可使图形水平、垂直等比例显示。

求解：单击，这时显示的解为对应第 1 特征值 Lambda(1)=19.7902 的特征函数 $u_{1,1}$ 的图形，如图 3-14。如果要输出小于 100 的特征值及对应的特征函数，只要单击 Solve 菜单中 Export Solution...选项，在打开的 Export 对话框中键入 u l (u 为特征函数变量，l 为特征值变量，注意各变量之间要用空格分开)，单击 OK 按钮。再在 MATLAB 命令窗口中键入 u，按回车键，立即显示按节点编号的特征函数值。如果输入 l，按回车键，则显示特征值：19.7902, 49.6518, 49.6602, 79.7728, 99.9254, 99.9951。它们分别为 $\pi^2(1^2+1^2)$，$\pi^2(1^2+2^2)$，$\pi^2(2^2+1^2)$，$\pi^2(2^2+2^2)$ 等的近似值。如果要作第 2 个、第 3 个……特征值对应的特征函数图形，只要在 Plot Selection 对话框的右下角的 Eigenvalue 项中选定某一特征值，单击 Plot 按钮，即可得到关于这个特征值对应的特征函数的图形。

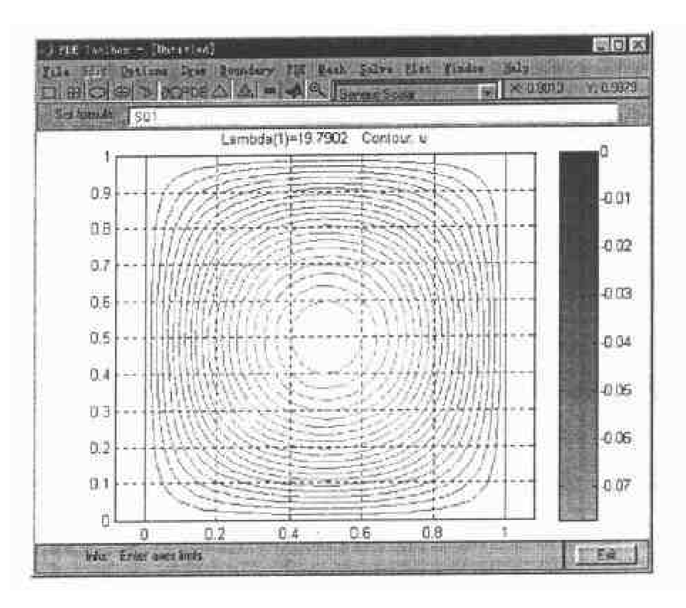


图 3-14

我们将上例中正方形区域用一L形多边形替代，其顶点坐标为(0,0), (−1,0), (−1,−1), (1,−1), (1,1)和(0,1)。操作步骤均与上例类似，打开 Plot Selection 对话框，选 Contour 项，单击 Plot 按钮，生成这一区域齐次边界条件下的特征值问题的解，如图 3-15。这时在[0 100]中有 16 个特征值，再次打开 Plot Selection 对话框，可在 Eigenvalue 栏的弹出式按钮中看到所有特征值，从中取定某个特征值，单击 Plot 按钮，可得到关于这个特征值对应的特征函数的图形。

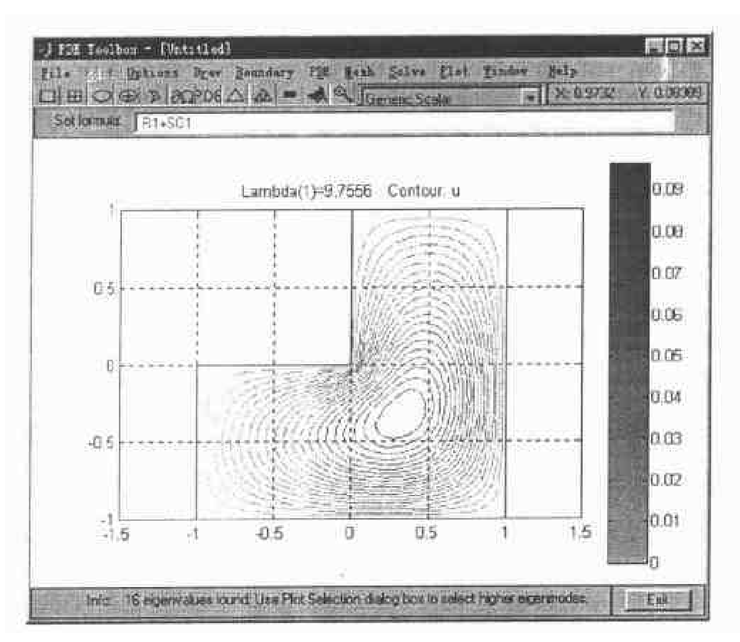


图 3-15

MATLAB 程序：

```
[p,e,t]=initmesh('lshapeg');
[p,e,t]=refinemesh('lshapeg',p,e,t);
[p,e,t]=refinemesh('lshapeg',p,e,t);
[v,l]=pdeeig('lshapeb',p,e,t,1,0,1,[0 100]);  %求解
pdesurf(p,t,v(:,1))  %绘图
figure
membrane(1,20,9,9)
figure
pdesurf(p,t,v(:,12))  %绘图
figure
membrane(12,20,9,9)
```

下面再举一个方域上混合边界条件的特征值问题：

$$\begin{cases} -\Delta u = \lambda u, & (x, y) \in \Omega = \{(x, y) \mid -1 < x, y < 1\}, \\ u|_{x=-1} = 0, \\ \left.\dfrac{\partial u}{\partial n}\right|_{y=\pm 1} = 0, \\ \left.\left(\dfrac{\partial u}{\partial n} - \dfrac{3}{4}u\right)\right|_{x=1} = 0, \end{cases}$$

类似前面的例子，首先作区域：方形的 4 个顶点为(−1,1), (−1, −1), (1, −1), (1,1)，在 Object Dialog 对话框中输入 Left 为-1, Bottom 为-1, Width 为 2, Height 为 2。

单击[图标]，逐段双击边界设置边界条件：左边界选 Dirichlet 条件，输入 h 为 1, r 为 0；右边界选 Neumann 条件，输入 q 为-3/4, g 为 0；上、下边界选 Neumann 条件，输入 q 为 0, g 为 0。

在 PDE Specification 对话框中设置方程类型为 Eigenmodes (特征值模式)，PDE 的系数设置为 c=1, a=0, d=1。

由于右边界上 Neumann 条件，特征值可以出现负值，因此在求小于 10 的特征值时应在 Solve Parameters 对话框中键入[-inf 10]。

作网格剖分，然后单击[图标]，得到第 1 个特征值对应的特征函数图形，如图 3-16。

再在 Plot Selection 对话框的 Eigenvalue 项中选取不同的特征值，比如第 2 个特征值 “2.079”，并选 Color, Contour, Height (3-D plot)和 Show mesh 四项，如图 3-17，可作出第 2 个特征值对应的特征函数的三维图形，如图 3-18。

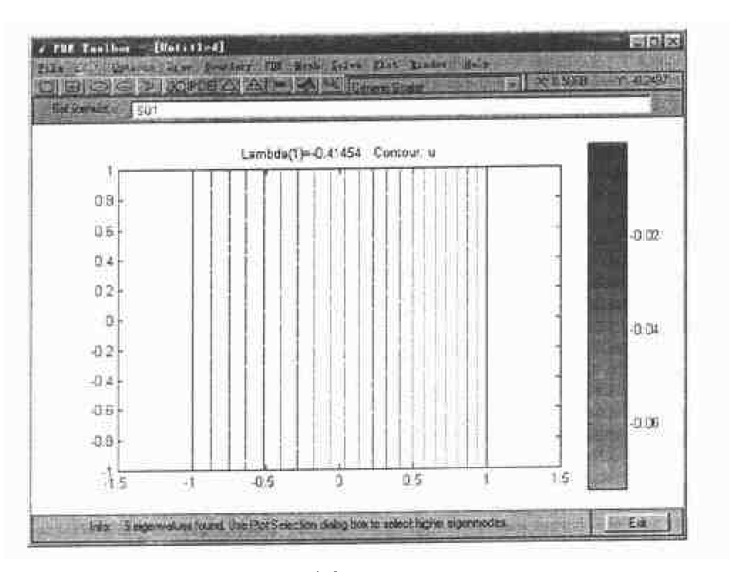


图 3-16

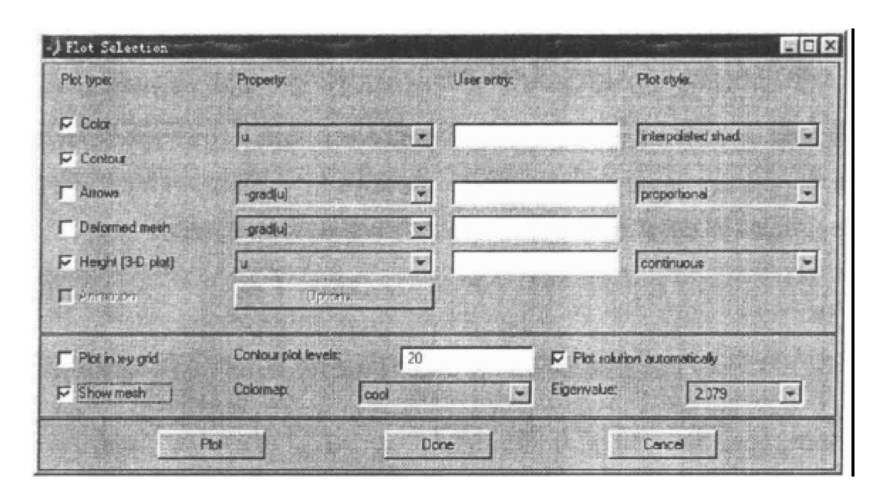


图 3-17

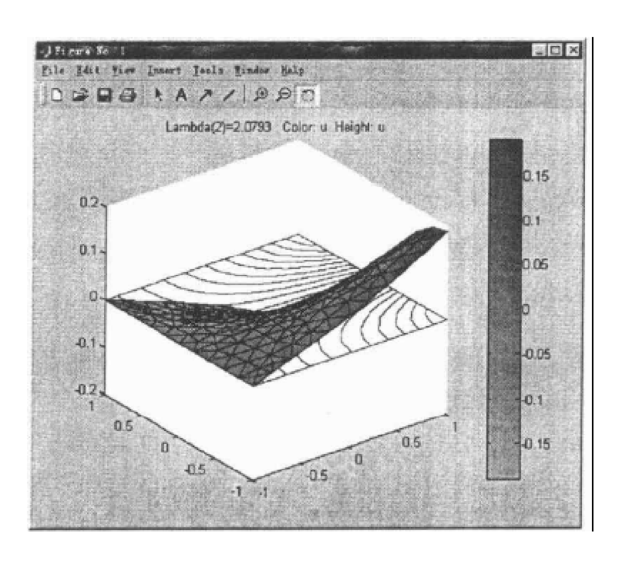


图 3-18

MATLAB 程序：

```
[p,e,t]=initmesh('squareg');
[p,e,t]=refinemesh('squareg',p,e,t);
[v,l]=pdeeig('squareb2',p,e,t,1,0,1,[-Inf 10]);
pdesurf(p,t,v(:,4))
```

3.5　应用模型

本节介绍 PDE Tool GUI 中应用模型的几个例子。

首先从 GUI 的右上方 Pop-up 菜单中选定应用模型，或者从 Options 菜单的 Application 选项的下一级子菜单中进行选择。

3.5.1 结构力学——平面应力问题

在结构力学中，有关应力和应变方程是从介质的力的平衡中导出的。平面应力问题通常考虑 xy 平面的平板问题，荷载仅在平面上，而 z 方向没有约束。

假设介质是各向同性和等温的，应力-应变关系可写成

$$\begin{pmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{pmatrix}=\frac{E}{1-\nu^2}\begin{pmatrix}1 & \nu & 0\\ \nu & 1 & 0\\ 0 & 0 & (1-\nu)/2\end{pmatrix}\begin{pmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{pmatrix}$$

其中，σ_x和σ_y是沿着 x 和 y 方向的应力，τ_{xy}是剪切应力。介质的性质是由弹性模量或称杨氏模量 E 以及 Poisson 比ν刻画的。

应变与 x 和 y 方向的位移 u 和 v 之间的关系为

$$\varepsilon_x=\frac{\partial u}{\partial x},\quad \varepsilon_y=\frac{\partial v}{\partial y},\quad \gamma_{xy}=\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}.$$

力的平衡方程为

$$-\frac{\partial \sigma_x}{\partial x}-\frac{\partial \tau_{xy}}{\partial y}=K_x,$$
$$-\frac{\partial \tau_{xy}}{\partial x}-\frac{\partial \sigma_y}{\partial y}=K_y.$$

其中 K_x 和 K_y 是体力。

综合上述关系，得到位移方程：

$$-\nabla\cdot(\underline{c}\otimes\nabla \boldsymbol{u})=\underline{\boldsymbol{K}},$$

其中 $\boldsymbol{u}$ 是二维向量，$\underline{c}$ 是秩为 4 的张量，可以写成 4 个 2×2 的矩阵 c_{11}，c_{12}，c_{21} 和 c_{22}。

$$c_{11}=\begin{pmatrix}2G+\mu & 0\\ 0 & G\end{pmatrix},\quad c_{12}=\begin{pmatrix}0 & \mu\\ G & 0\end{pmatrix},$$
$$c_{21}=\begin{pmatrix}0 & G\\ \mu & 0\end{pmatrix},\quad c_{22}=\begin{pmatrix}G & 0\\ 0 & 2G+\mu\end{pmatrix},$$

其中剪切模 G 定义为$G=\dfrac{E}{2(1+\nu)}$，μ定义为$\mu=2G\dfrac{\nu}{1-\nu}$。体力

$$\underline{\boldsymbol{K}}=\begin{pmatrix}K_x\\ K_y\end{pmatrix}.$$

这是一个椭圆型方程组。只要在 PDE Tool GUI 中选择应用模型 Structural Mech., Plane Stress，并在 PDE Specification 对话框中键入介质参数 E 和ν (nu) 以及体力 $\underline{\boldsymbol{K}}$的取值，即可求解这类问题。

在这个应用模型中，还可以解如下特征值问题：

$$-\nabla\cdot(\underline{c}\otimes\nabla u)=\lambda \underline{d}u,$$

$$\underline{d}=\begin{pmatrix}\rho & 0\\ 0 & \rho\end{pmatrix}.$$

密度值 ρ 可以利用 PDE Specification 对话框输入。在 Plot Selection 对话框中，可以用彩色可视化表示 x 和 y 方向的位移 u 和 v、位移向量(u,v)的绝对值以及等值线，还可以用矢量图或变网格图表示位移向量场(u, v)。另外，为了作等值线的彩色可视化图形，可以从以下 15 个张量中选择：

$\mathrm{ux}=\dfrac{\partial u}{\partial x}$，　$\mathrm{uy}=\dfrac{\partial u}{\partial y}$，　$\mathrm{vx}=\dfrac{\partial v}{\partial x}$，　$\mathrm{vy}=\dfrac{\partial v}{\partial y}$，

exx——x 方向的应变 ε_x，

eyy——y 方向的应变 ε_y，

exy——切应变 γ_{xy}，

sxx——x 方向的应力 σ_x，

syy——y 方向的应力 σ_y，

sxy——剪切应力 τ_{xy}，

e1——第一主应变 ε_1，

e2——第二主应变 ε_2，

s1——第一主应力 σ_1，

s2——第二主应力 σ_2，

von Mises——von Mises 等效应力 ($\sqrt{\sigma_1^2+\sigma_2^2-\sigma_1\sigma_2}$).

【例 3.5-1】　考虑一块钢板，其左下方的直角断面被镶住，其右上方的圆弧断面上受拉力，其他边均自由。假设钢板面为 1 m×1 m，厚为 1 mm，镶住的边为 $\frac{1}{3}$m×$\frac{1}{3}$m，右上方的圆弧的圆心为$(\frac{2}{3},\frac{2}{3})$，半径为 $\frac{1}{3}$，圆弧从$(\frac{2}{3},1)$到$(1,\frac{2}{3})$。杨氏模量为 196×10^3 (MN/m^2)，Poisson 比为 0.31。圆弧边界受外法向荷载为 500 N/m，钢板表面牵引力为 0.5 MN/m^2，其中力的单位为 MN。

我们感兴趣的是计算 x 和 y 方向的应变和应力、剪切应力以及 von Mises 等效应力。

使用 GUI 求解。打开 PDE Toolbox GUI 图形用户界面，选择 Structural Mech., Plane Stress 应用模式。

(i) **区域设置**　单击 Options 菜单中 Axes Limits 选项，键入 X-axis range：[0 1.2]，Y-axis range：[0 1.2]，单击 Apply 按钮。再选择 Options 菜单中 Axes Equal

命令。

单击 Draw 菜单中 Polygon 选项，绘制七边形 P1，双击 P1，键入角点坐标为(0,1),(0,1/3),(1/3,1/3) (1/3,0),(1, 0),(1,2/3),(2/3,1)。再选 Draw 菜单中 Ellipse/circle (centered)选项，拖动鼠标右键画圆，双击圆，打开 Object Dialog 对话框，键入圆的圆心为(2/3,2/3)，半径为 1/3。如图 3-19。

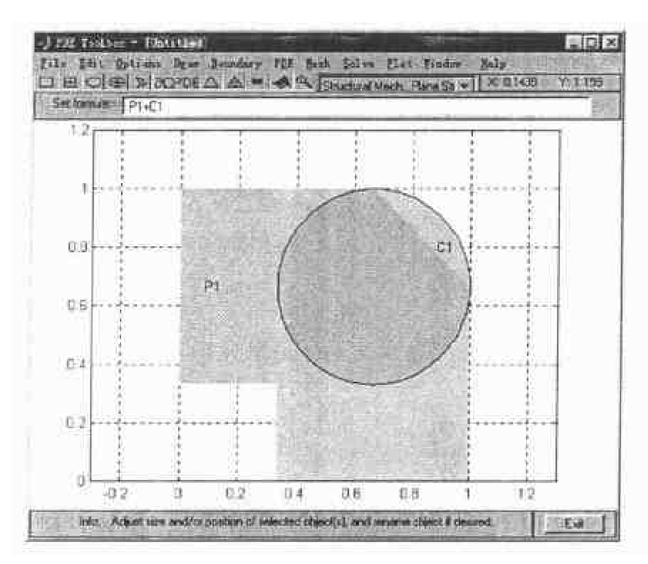


图 3-19

选择 Boundary 菜单中 Boundary Mode 命令，进入边界模式。单击 Boundary 菜单中 Remove All Subdomain Borders 选项，去除子域边界。

(ii) **设置边界条件** 逐段双击边界设置边界条件：左下方的直角边上是固定的，即满足齐次 Dirichlet 条件，h11=h22=1, h12=h21=0, r1=r2=0；右上方圆弧边上满足 Neumann 条件，qij=0 (i,j=1,2), g1=0.5*nx, g2=0.5*ny；其余边界是自由的，即满足齐次 Neumann 条件：qij=0 (i,j=1,2), g1=g2=0。

(iii) **设置方程类型** 打开 PDE Specification 对话框，键入参数：杨氏模量 E=196E3，Poisson 比 nu=0.31，因为没有体力，故 Kx 和 Ky 为 0，而 ρ (rho) 在本模型中不使用，可不用管。

(iv) **网格剖分** 单击█，再单击█加密剖分。

(v) **求解** 单击 Plot 菜单中 Parameters...选项，打开 Plot Selection 对话框，设置输出各种变量的图形，如位移、应变、应力、剪切应力、剪切应变、von Mises 等效应力等等。比如，利用变形网格可以作出 von Mises 等效应力和位移图形，在 Plot Selection 对话框中选择 Color, Contour, Deformed mesh 和 Show mesh 四项、在 Color 栏的 Property (属性)列中选择 von Mises stress，单击 Plot 按钮，显示出 von Mises 等效应力和位移图形，如图 3-20。据此可以分析应力集中等性质。

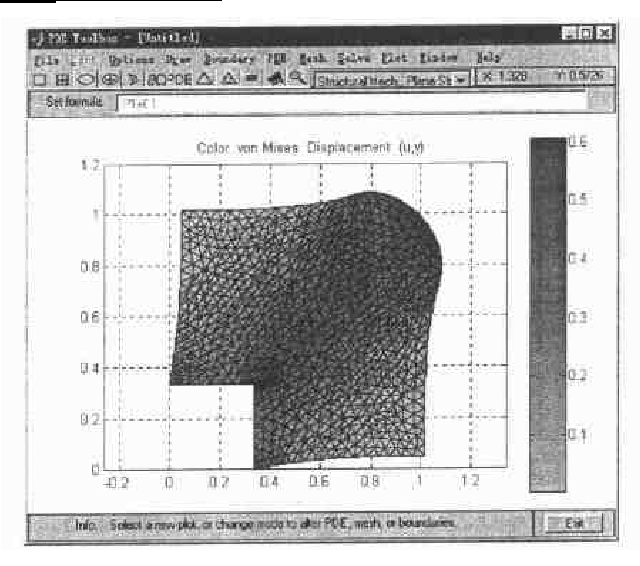


图 3-20

3.5.2　结构力学——平面应变问题

现在考虑平面应变问题，即除了 x 和 y 方向外，z 方向没有位移。首先在 PDE Toolbox 窗口选 Structured Mech., Plane Strain 应用模型。平面应力关系与平面应变有所不同，但是材料的参数仍可以使用。平面应变方程与平面应力方程的区别在于：

在张量 $\underline{c}$ 中的参数 μ 定义为 $\mu = 2G\dfrac{v}{1-2v}$。

von Mises 等效应力计算公式为

$$\sqrt{(\sigma_1^{\ 2}+\sigma_2^{\ 2})(v^2-v+1)-\sigma_1\sigma_2(2v^2-2v-1)}.$$

3.5.3　静电场问题

这一应用模型包括高压设备、电子仪器和电容器等实际问题。所谓“静”是指场对于时间的变化很缓慢，波长要比所考虑的区域尺度大得多。在静电学中，电位 V 与场强 $\boldsymbol{E}$ 的关系为 $\boldsymbol{E}=-\nabla V$，考虑在各向同性介质中 $\boldsymbol{D}=\varepsilon\boldsymbol{E}$，由 Maxwell 方程知，电位移矢量 $\boldsymbol{D}$ 满足 $\nabla\cdot\boldsymbol{D}=\rho$，于是得到关于电位 V 的 Poisson 方程：

$$-\nabla\cdot(\varepsilon\nabla V)=\rho,$$

其中 ε 是介电系数，ρ 是空间电荷密度。这里 ε 实际上可写成 $\varepsilon\varepsilon_0$，ε_0 是真空介电系数(8.854×10^{-12}法拉第/米)，ε 是相对介电系数，对不同介电材料其值不同，例如空气为 1.00059，变压器油为 2.24 等等。

使用 PDE Toolbox 应用模型中的 Electrostatics 便可求解上述方程。在 PDE Specification 对话框中键入 ε 和 ρ 的值，边界条件分为 Dirichlet 条件和 Neumann 条件，前者在边界上给出电位值 V，后者在边界上给出电荷密度 $\boldsymbol{n}\cdot(\varepsilon\nabla V)$。解的可视化包括电位势 V、电场强度 $\boldsymbol{E}$ 和电位移矢量 $\boldsymbol{D}$。

考虑一个正方形构件的静电势问题，它的外边界边长为 0.5，正方形内孔边长为 0.2，在内边界上电势保持在 1000 V，而在外边界上电势保持在 0 V，它们均为 Dirichlet 条件。在区域内不含电荷，于是导出 Laplace 方程 $\Delta V=0$。

下面便用 GUI 求解。首先选择应用模型为 Electrostatics。作中心在原点边长为 0.2 的正方形 SQ1，在 Object Dialog 对话框中设置 Left=-0.1，Bottom=-0.1，Width=0.2，Height=0.2；再作中心在原点、边长为 0.5 的正方形 SQ2，在 Object Dialog 对话框中设置 Left=-0.25，Bottom=-0.25，Width=0.5，Height=0.5。在 Set formula 栏中键入 SQ2-SQ1。定义边界条件：双击内边界选 Dirichlet 条件，h=1，r=1000，外边界也设定为 Dirichlet 条件，h=1，r=0。再打开 PDE Specification 对话框，设置电荷密度(rho)为 0，介电系数(epsilon)为 1 (由于 Laplace 方程是齐次的，所以介电系数值不影响结果)。再生成网格，单击▦，便可解出方程。如果在 Solve Parameters 对话框中使用 Adaptive mode，使可以在角点这种梯度大的区域加密网格，以提高解的精度，例如设置 Maximum number of triangles (最大三角形数)为 500。单击 Mesh 菜单下 Refine Mesh 命令，便得网格均匀加密。最后单击▦，便得到解的图形。如果要作等势线图形，在 Plot Selection 对话框中选 Contour，设置 Contour plot levels 为 10，显示每 100 V 一条等势线，从 0 到 1000 共 10 条等势线，如图 3-21。

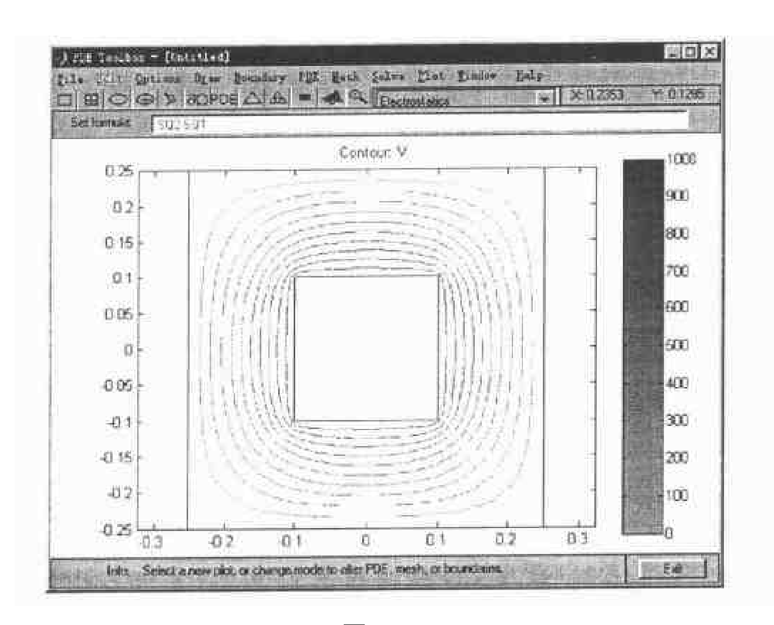


图 3-21

3.5.4　静磁场问题

在磁铁、电动机、变压器这样一类问题中都有静磁场问题，所谓“静”是指关于时间的变化缓慢，因而考虑如下定常的 Maxwell 方程：

$$\nabla\times\boldsymbol{H}=\boldsymbol{J},$$
$$\nabla\cdot\boldsymbol{B}=0,$$
$$\boldsymbol{B}=\mu\boldsymbol{H},$$

其中 $\boldsymbol{B}$ 是磁感应强度，$\boldsymbol{H}$ 是磁场强度，$\boldsymbol{J}$ 是电流密度，μ 是材料的磁导率。由于 $\nabla\cdot\boldsymbol{B}=0$，故存在一个静磁矢势 $\boldsymbol{A}$，使得

$$\boldsymbol{B}=\nabla\times\boldsymbol{A},$$

及

$$\nabla\times(\frac{1}{\mu}\nabla\times\boldsymbol{A})=\boldsymbol{J}.$$

平面问题中假设电流平行于 z 轴，故 $\boldsymbol{A}$ 仅有 z 分量

$$\boldsymbol{A}=(0,0,A),\quad \boldsymbol{J}=(0,0,J),$$

从而，上述方程可以简化成椭圆型 PDE:

$$-\nabla\cdot(\frac{1}{\mu}\nabla A)=J,$$

其中 $J=J(x,y)$。

对于二维情况，可以计算磁感应强度为

$$\boldsymbol{B}=(\frac{\partial A}{\partial y},-\frac{\partial A}{\partial x},0),$$

磁场强度为

$$\boldsymbol{H}=\frac{1}{\mu}\boldsymbol{B}.$$

对于不同性质的材料组成的区域之间的交界面上，$\boldsymbol{H}\times\boldsymbol{n}$ 是连续的，也就是 $\frac{1}{\mu}\frac{\partial A}{\partial n}$ 连续。由于使用有限元法解 PDE 问题，从变分问题出发，故交界面上这一条件作为自然边界条件处理。

在强磁性材料中，μ 依赖于场 $|\boldsymbol{B}|=|\nabla A|$，从而导致非线性解。

如果是 Dirichlet 边界条件，则要求在边界上给出磁势 A 的值。如果是 Neumann 边界条件，则要求在边界上给出 $\boldsymbol{n}\cdot(\frac{1}{\mu}\nabla A)$ 的值，也就是在边界上给出磁场强度 $\boldsymbol{H}$ 的切向分量。磁势 A、磁场强度 $\boldsymbol{H}$ 以及磁感应强度 $\boldsymbol{B}$ 的可视化均可实现，还可以作 $\boldsymbol{B}$ 和 $\boldsymbol{H}$ 的向量场图形。

实例

考虑由双极电动机锭子线圈产生的静磁场问题。假设马达很长，使得端点的影响可以忽略不计，从而简化成二维模型处理。

区域由四部分组成：

- 两个铁磁体：定子和转子
- 定子和转子之间的间隙
- 带有直流的转子线圈

在空气和线圈中磁导率 μ 为 1，而定子和转子的磁导率为

$$\mu = \frac{\mu_{max}}{1+c\|\nabla(A)\|^2} + \mu_{min} ,$$

其中 $\mu_{max} = 5000$，$\mu_{min} = 200$，$c = 0.05$，这些是由变压器钢材决定的。

- 线圈的电流密度 J 为 1，其他均为 0。

这一问题的磁势 A 关于 y 轴是对称的，关于 x 轴是反对称的，从而只需考虑 $x \geq 0$ 和 $y \geq 0$ 部分，而且在 x 轴上满足 Neumann 条件：$\boldsymbol{n}\cdot(\frac{1}{\mu}\nabla A) = 0$，在 y 轴上满足 Dirichlet 条件：$A = 0$。由于马达外面的磁场可以忽略，故外边界均为 Dirichlet 条件：$A = 0$。

这一问题比较复杂，它的区域包括 5 个圆和 2 个矩形，它的方程在不同区域也不完全相同，而且包括非线性方程。下面详细介绍利用 GUI 求解的过程首先在 PDE Tool 窗口应用模型中选定 Magnetostatics，然后单击 Solve 菜单中 Parameters...选项，打开 Solve Parameters 对话框，选定 Use nonlinear solver。

(i) **区域设置** 单击 Options 菜单中 Grid Spacing...选项，打开 Grid Spacing 对话框，取消 Auto 选项，键入 X-axis linear spacing：-1.5:0.5:1.5 (调整栅格，步长为 0.5)；Y-axis linear spacing：-1:0.2:1。单击 Options 菜单中 Axes Limits...选项，打开 Axes Limits 对话框，取消 Auto 选项，键入 X-axis range：[-1.5 1.5]，Y-axis range：[-1 1]，完成坐标轴的显示设置。键入如下命令函数画圆和矩形：

```
pdecirc(0,0,1,'C1')
pdecirc(0,0,0.8,'C2')
pdecirc(0,0,0.6,'C3')
pdecirc(0,0,0.5,'C4')
pdecirc(0,0,0.4,'C5')
pderect([-0.2 0.2 0.2 0.9],'R1')
pderect([-0.1 0.1 0.2 0.9],'R2')
pderect([0 1 0 1],'SQ1')
```

以上所绘制的圆 C1,C2,C3,C4,C5 的圆心均为(0,0)，半径分别为 1,0.8,0.6,0.5,0.4。矩形 R1,R2 的高为 0.7，宽分别 0.4,0.2。正方形 SQ1 的边长为 1。在 Set formula 栏中键入(C1+C2+C3+C4+C5+R1+R2)*SQ1。

单击 Boundary 菜单下 Boundary Mode 选项，进入边界模式，如图 3-22。然后删去图中的部分线段和弧段，具体作法如下：单击所要删去的线段或弧段，执行 Boundary 菜单中 Remove Subdomain Border 命令，所选的线段或弧段即被删除。再选择 Boundary 菜单中 Show Subdomain Labels 命令，显示区域编号，如图 3-23。

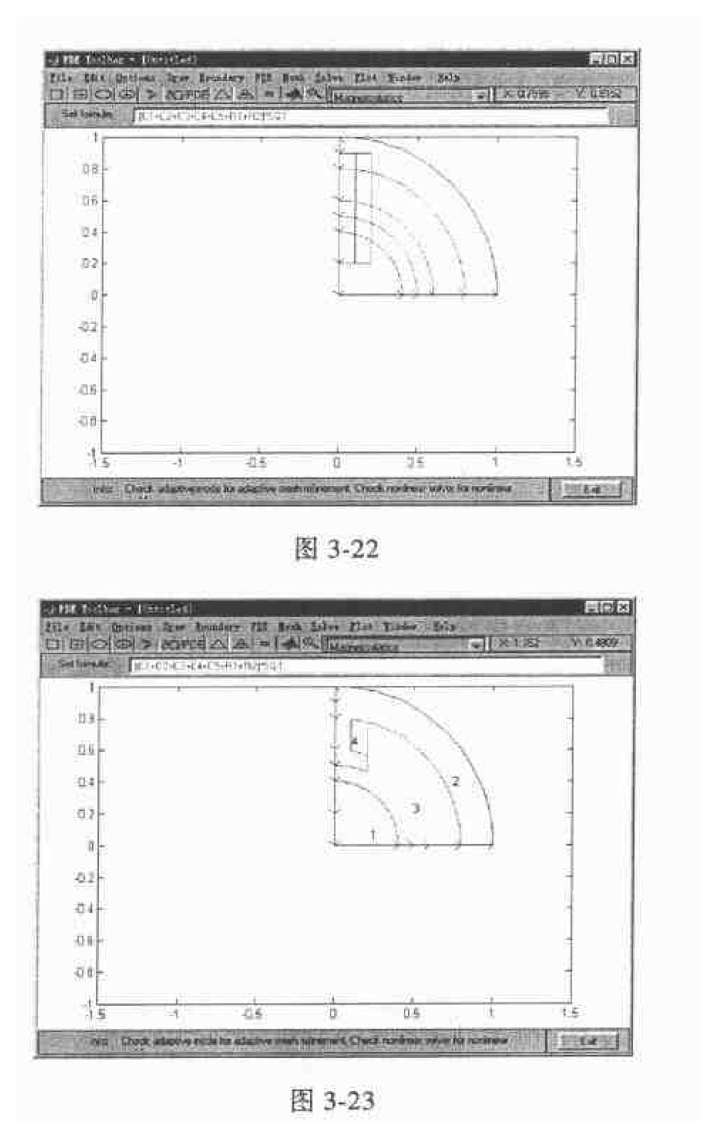


图 3-22

图 3-23

重新调整坐标轴的范围，便于观察图形的显示。单击 Options 菜单中 Axes Limits...选项，打开 Axes Limits 对话框，取消 Auto 选项，键入 X-axis range：[-0.2 1.2]，Y-axis range：[-0.2 1.2]。再单击 Options 菜单中 Axes Equal 选项，

使两坐标轴的尺寸比例一致。

(ii) **边界条件设置** 逐段双击边界设置边界条件：X 轴边界选齐次 Neumann 条件、键入 $g=0$ ， $q=0$ 。其他边界选齐次 Dirichlet 条件，已默认这种边界条件。

(iii) **方程的设置** 选择 PDE 菜单中 PDE Mode 命令，进入 PDE 模式。分别双击 4 个不同区域，设置方程的系数：线圈区域 4 键入 mu=1，J=1；定子区域 2 和转子区域 1 键入 mu=5000./(1+0.05*(ux.^2+uy.^2))+200，J=0，这里系数含有非线性；空隙区域 3 键入 mu=1，J=0。

(iv) **求解** 在 Plot Selection 对话框中选定彩色 Color、等值线 Contour、矢量线 Arrows，设置 Contour plot levels：10，如图 3-24。单击 Plot 按钮，可以立即求出这些量，如图 3-25。如果选 Height (3-D plot)，可得到如图 3-26 (1)所示的三维解的彩色图形。在三维图形中，可以任意拖动鼠标旋转三维坐标轴，获得不同角度的三维彩色图形，如图 3-26 (2)。

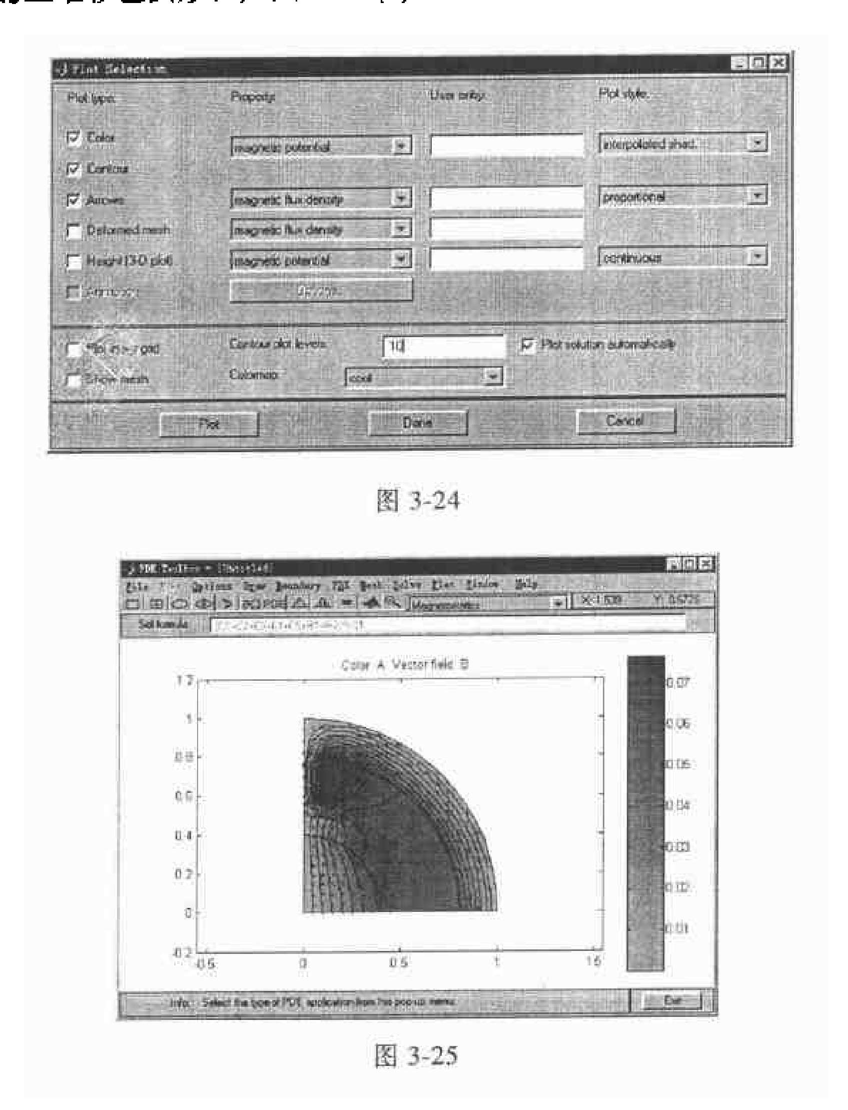
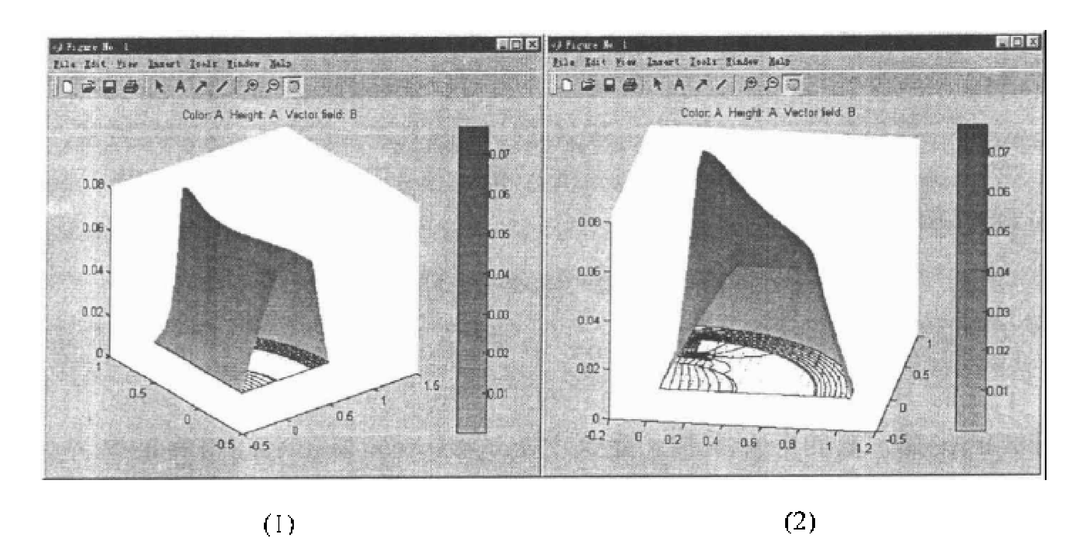


图 3-24

图 3-25



(1)　　　　(2)

图 3-26

3.5.5 交流电磁场问题

在研究交流电的马达、变压器和导体问题时会遇到这一问题。考虑均匀电介质，介电系数为 ε 、磁导率为 μ ，不带任何电荷，这时电磁场服从一般的 Maxwell 方程组：

$$\nabla\times \boldsymbol{E} = -\mu\frac{\partial \boldsymbol{H}}{\partial t},$$

$$\nabla\times \boldsymbol{H} = \varepsilon\frac{\partial \boldsymbol{E}}{\partial t} + \boldsymbol{J}.$$

在没有电流的情况下，可以从第一式消去 $\boldsymbol{H}$ ，第二式消去 $\boldsymbol{E}$ ，于是它们满足波速为 $\sqrt{\varepsilon\mu}$ 的波动方程：

$$\Delta \boldsymbol{E} - \varepsilon\mu\frac{\partial^2 \boldsymbol{E}}{\partial t^2} = 0, \quad \Delta \boldsymbol{H} - \varepsilon\mu\frac{\partial^2 \boldsymbol{H}}{\partial t^2} = 0.$$

进一步研究无电荷的均匀电介质情况，这时电介质系数为 ε ，磁导率为 μ ，电导率为 σ ，于是电流密度为

$$\boldsymbol{J} = \sigma \boldsymbol{E}.$$

这样就得到有阻尼的波动方程：

$$\Delta \boldsymbol{E} - \mu\sigma\frac{\partial \boldsymbol{E}}{\partial t} - \varepsilon\mu\frac{\partial^2 \boldsymbol{E}}{\partial t^2} = 0.$$

关于 $\boldsymbol{H}$ 也有类似的有阻尼的波动方程。

考虑频率一定的谐波情形，只需用复指数 $\boldsymbol{E}_c e^{j\omega t}$ 代替上述式中的 $\boldsymbol{E}$ ($j=\sqrt{-1}$)，对于 PDE Toolbox 模型中的平面情形有

$$\boldsymbol{E}_c = (0,0,E_c),\ \ \boldsymbol{J} = (0,0,Je^{j\omega t}),$$

以及磁场

$$\boldsymbol{H} = (H_x, H_y, 0) = -\frac{1}{j\mu\omega}\nabla\times\boldsymbol{E}_c,$$

标量 E_c 的方程为

$$-\nabla\cdot(\frac{1}{\mu}\nabla E_c) + (j\omega\sigma - \omega^2\varepsilon)E_c = 0.$$

解这个方程需要用 PDE Toolbox 中 AC Power Electromagnetics 模型。这是一个复的 Helmholtz 方程，它描述在非理想介质和良导体 $(\sigma \gg \omega\varepsilon)$ 中平面电磁波的传播。复的电介系数 ε_c 定义为 $\varepsilon_c = \varepsilon - j\sigma/\omega$。在不同材料的交界面上 ε 和 μ 是不连续的，由于满足变分原理的自然边界条件，故不需要单独表示出来。

关于 PDE 参数需要在 PDE Specification 对话框中输入圆频率 ω、磁导率 μ、电导率 σ 以及电介质系数 ε。

这个模型的边界分 Dirichlet 条件和 Neumann 条件，后者给的是 E_c 的法向导数，也就是 $\boldsymbol{H}$ 的切向分量：

$$H_t = \frac{j}{\omega}\boldsymbol{n}\cdot(\frac{1}{\mu}\nabla E_c).$$

当解出电磁强度 $\boldsymbol{E}$，可进一步计算电流密度 $\boldsymbol{J} = \sigma\boldsymbol{E}$ 以及磁感应强度 $\boldsymbol{B} = \frac{j}{\omega}\nabla\times\boldsymbol{E}$，还可以作出场强 $\boldsymbol{E}$、电流密度 $\boldsymbol{J}$、磁场强度 $\boldsymbol{H}$ 和磁感强度 $\boldsymbol{B}$ 的图形。同时热阻率 $Q = E_c^2/\sigma$ 也可以作图，对于 $\boldsymbol{H}$ 和 $\boldsymbol{B}$ 也可以用矢量作出其向量场。

实例

考虑圆截面线圈通过交流电流，这时会产生趋肤效应。铜的电导率 σ 为 57×10^6，磁导率为 1，即 $\mu = 4\pi\times10^{-7}$，在线频率为 50 Hz，$\omega^2\varepsilon$ 项可以忽略。

由于感应作用，导体内部的电流密度远远小于外表面的电流 $J_S = 1$，这里电场的 Dirichlet 条件为 $E_c = 1/\sigma$。这种情况能够求出解析解：

$$J = J_S\frac{J_0(kr)}{J_0(kR)},$$

其中 $k = \sqrt{j\omega\mu\sigma}$，$R$ 是线圈的半径，r 是离圆心的距离，$J_0(x)$ 是第一类零阶 Bessel 函数。

使用 GUI 求解。在 PDE Toolbox GUI 窗口中，选择 AC Power Electromagnetics 应用模型。

(i) **画图：** 单击工具栏中按钮，在 GUI 中用鼠标右键拖拉出圆，双击圆，输入圆心坐标 X-center 为 0, Y-center 为 0，半径 Radius 为 0.1。

(ii) **设置边界条件**：单击 Boundary 菜单中 Boundary Mode 选项，再单击 Boundary 菜单中 Specify Boundary Conditions...，打开 Boundary Condition 对话框，选 Dirichlet 条件，h=1，r=1/57*1E6。

(iii) **设置方程类型**：打开 PDE Specification 对话框，选 Elliptic，键入参数：Omega=2*pi*50, mu=4*pi*1E-7, sigma=57E6, epsilon=8.8E-12。

(iv) **生成网格**：单击网络剖分，再单击加密网格。

(v) **作图**：打开 Plot Selection 对话框，选 Color、Contour、Height (3-D plot) 及 Show mesh 四项，单击 Plot 按钮，即显示出解的三维彩色图形。解是复数，但显示的只是实数部分的解，如图 3-27。

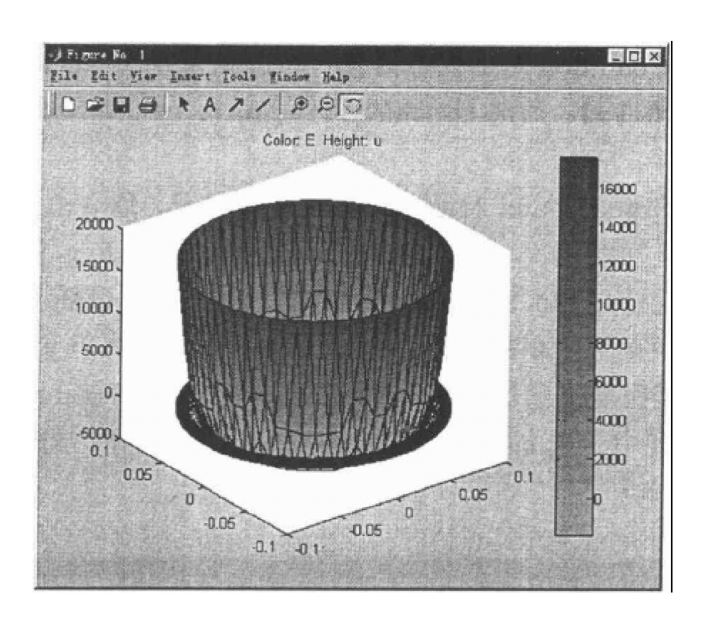


图 3-27

3.5.6　直流导电介质问题

考虑导电介质的电导率为 σ，在定常电流下，由电流密度 $\boldsymbol{J}$ 与电场强度 $\boldsymbol{E}$ 的关系 $\boldsymbol{J}=\sigma\boldsymbol{E}$，以及 $\nabla\cdot\boldsymbol{J}=Q$（$Q$ 为电流源），可得到用电位 V 表示的 Poisson 方程:

$$-\nabla\cdot(\sigma\nabla V)=Q.$$

如果边界给 Dirichlet 条件，即是在边界给出电位 V 的值；如果边界给 Neumann 条件，即是在边界给出电流密度的法向分量 $(\boldsymbol{n}\cdot(\sigma\nabla(V)))$。也可以给出一般 Neumann 条件为

$$\boldsymbol{n}\cdot(\sigma\nabla(V))+qV=g.$$

在 PDE Toolbox 中可以作电位 V、电场强度 $\boldsymbol{E}$ 以及电流密度 $\boldsymbol{J}$ 的图形。当 σ

是各向同性的时，等电位线与电流方向处处垂直。

例 在一个矩形介质中有两个圆，这两个圆均由金属导体组成，其中一个圆保持电位为 1，另一个保持电位为−1，而矩形外边界满足 Neumann 条件 $\frac{\partial V}{\partial n}=0$，介质的电导率为 1。

使用 GUI 求解。打开 PDE Tool GUI 图形用户界面窗口，在应用模型中选 Conductive Media DC，作矩形 R1，其 4 个顶点坐标为(−1.2, −0.6)，(1.2, −0.6)，(1.2,0.6)，(−1.2,0.6)。再作两个半径为 0.3 的圆 C1 和 C2 表示导体，其圆心分别为(−0.6,0)和(0.6,0)。在 Set formula 栏中输入：R1−(C1+C2)，完成区域的设置。

在 Boundary Condition 对话框中设置：外边界为齐次 Neumann 条件，g=0，q=0；左边圆的边界 V=1，即选 Dirichlet 条件，h=1，r=1；右边圆的边界 V=−1，即选 Dirichlet 条件，h=1，r=-1。

再打开 PDE Specification 对话框，设置方程类型，取 q=0，sigma=1。然后初始化网格，并且加密两次。还可以通过调整优化网格(Jiggle Mesh)，使三角剖分更合理。最后打开 Plot Selection 对话框，选 Color，Contour，单击 Plot 按钮，得到电位的等值线，如图 3-28。y 轴上的电位为 0，对于 y 轴点有反对称分布。还可以作电流密度 $\boldsymbol{J}$ 的等值线、矢量线。在 Plot Selection 对话框的 Color 栏对应的 Proper 列中，选 current density，单击 Plot 按钮，得到电流密度 $\boldsymbol{J}$ 的等值线，如图 3-29。

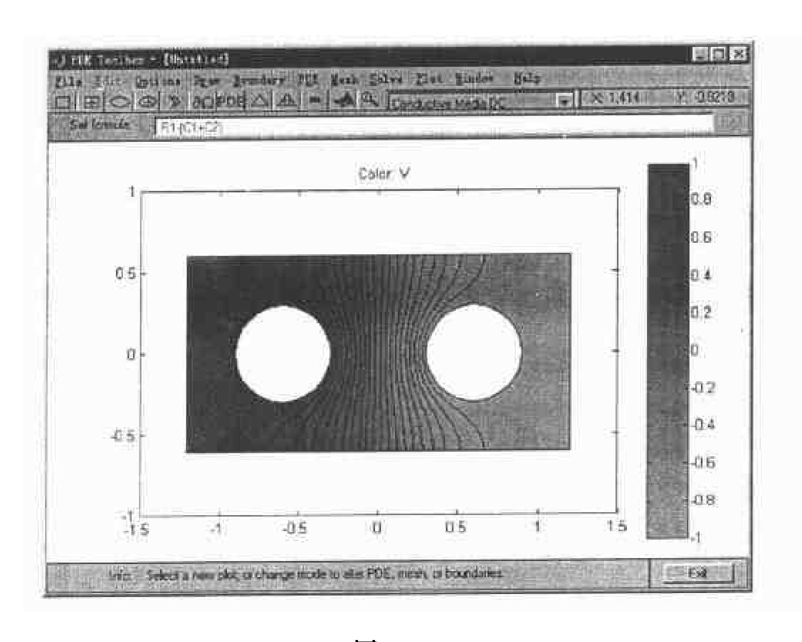


图 3-28

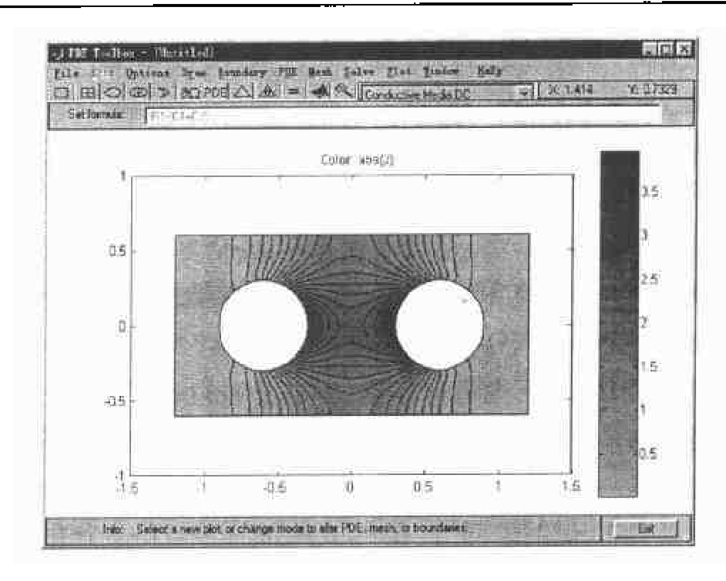


图 3-29

3.5.7 热的传输问题

这类问题属于抛物型偏微分方程:

$$\rho C\frac{\partial T}{\partial t}-\nabla\cdot(k\nabla T)=Q+h\cdot(T_{\text{ext}}-T).$$

它描述平面的热传输现象，以及由轴对称三维问题经过降维后的热传输问题，其中 T 为温度，其他参数为：ρ——密度，C——比热，k——导热系数，Q——热源，h——对流热的传输系数，T_{ext}——环境温度，$h\cdot(T_{\text{ext}}-T)$表示从环境向区域内部的传输热量。

如果仅考虑定态情况，那么方程变为

$$-\nabla\cdot(k\nabla T)=Q+h(T_{\text{ext}}-T).$$

边界条件仍然可以有 Dirichlet 条件、Neumann 条件和一般 Neumann 条件。

在 PDE Toolbox 中我们可以作温度、温度梯度、热流 $k\nabla T$ 的可视化图形。

例 不同介质系数的热的传输问题。

考虑一个方形区域，导热系数为 10，密度为 2。在方形区域内有一个菱形区域的热源为 4，导热系数为 2，密度为 1。两个区域的比热都为 0.1。

用 GUI 求解。打开 PDE Toolbox GUI 窗口,在应用模型中选 Heat Transfer。

设置坐标的显示和范围：单击 Options 菜单中 Grid Spacing...选项，在打开的 Grid Spacing 对话框中选 Auto (自动)；打开 Axes Limits 对话框，设置 X 轴和 Y 轴的范围为[0 3]，单击 Apply 按钮，再单击 Close 按钮关闭 Axes Limits 对话框。

绘制区域：设置方形区域的顶点为(0,0), (3,0), (3,3)和(0,3)；菱形区域的

顶点为(1.5,0.5), (2.5,1.5), (1.5,2.5)和(0.5,1.5)。单击 Options 菜单中 Axis Equal 选项，使两坐标轴的显示比例一致。

因为所有外边界的温度保持为 0，可知边界满足默认的齐次 Dirichlet 边界条件。

再在 PDE Mode 模式下，分别双击两个区域设置抛物型（Parabolic）PDE 参数：在方形区域中键入密度 rho 为 2，比热 C 为 0.1，导热系致 k 为 10，热源 Q 为 0。在菱形区域中键入密度 rho 为 1，比热 C 为 0.1，导热系数 k 为 2，热源 Q 为 4。导热项 $h(T_{ext}-T)$ 没有作用，故取 h=Text=0。

因为要求解随时间变化的 PDE，故需要定义初始值和时间间隔。打开 Solve Parameters 对话框。这一问题的动态变化非常快，温度在大约 0.1 单位时间即趋于定常。为了捕捉动态的有兴趣部分，需输入时间向量 Time 为 logspace(-2,-1,10), logspace(-2,-1,10)给出了 10^{-2}=0.01 和 10^{-1}=0.1 之间的 10 个对数空间致。

温度的动态性质可视化的最好方法就是作出解的动画。在 Plot Selection 对话框中选择 Color、Height(3-D plot)和 Animation，单击 Plot 按钮，即可显示解的动画过程。也可以选 Plot in x-y grid，使用矩形网格化代替三角形网格，以使有效地加快动画过程。如果在 Plot Selection 对话框中选择 Color、Contour 和 Arrows，单击 Plot 按钮，给出 Time=0.1 时温度 T 和矢量线-grad(T)的图形，如图 3-30。

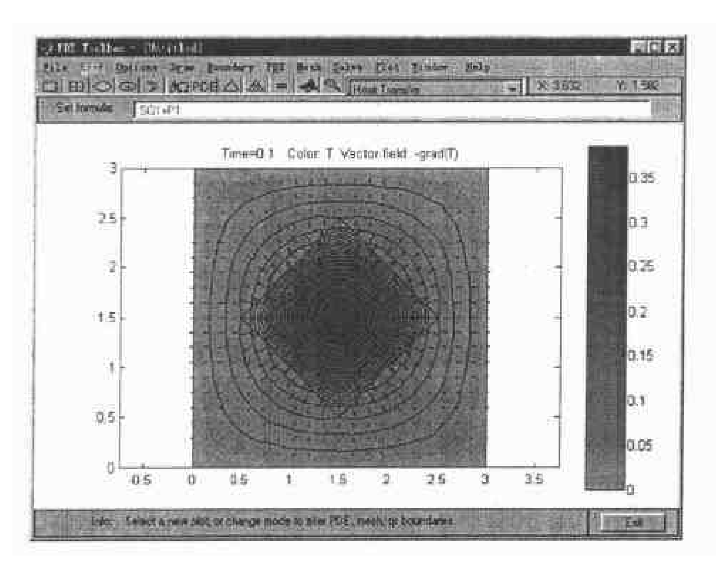


图 3-30

3.5.8 扩散问题

因为热的传输是一个扩散过程，因此一般的扩散方程也与热传导方程相同：

$$\frac{\partial c}{\partial t}-\nabla\cdot(D\nabla c)=Q,$$

其中 c 是浓度，D 是扩散系散，Q 是源项。

扩散过程可以是各向异性的，只要取 D 为 2×2 的矩阵。边界条件可以是 Dirichlet 条件，或者是 Neumann 条件。浓度、浓度梯度、热流的可视化可以在 PDE Tool 图形用户界面中由 Plot Selection 对话框中参数的设定来实现。

3.6 输出计算结果的例子

利用 MATLAB 图形用户界面求解 PDE 问题，定解区域可以是任意二维区域，网格剖分的节点及单元编号以及节点上的数值解，都可以十分方便地以数值方式输出，便于进一步研究。现在我们作一个十分复杂的图形，区域的边界可以用多边形绘图工具，或单击菜单 Draw 下的 Polygon 选项，画折线段逼近复杂弧线区域，如图 3-31。

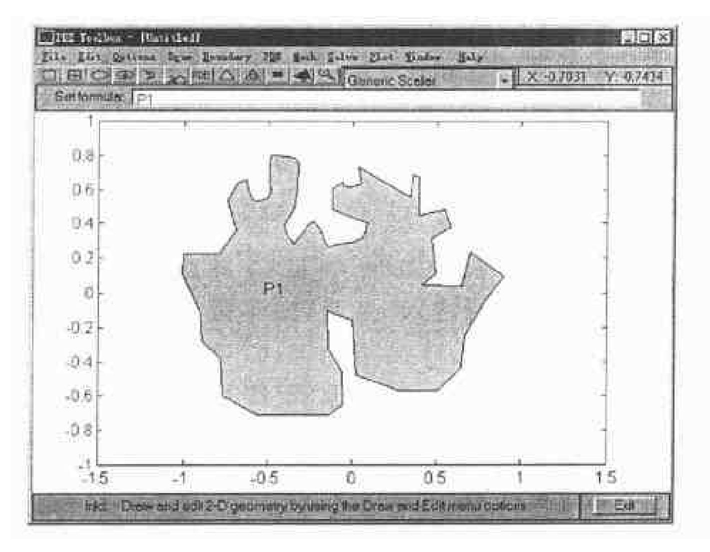


图 3-31

双击区域，打开 Object Dialog 对话框，如图 3-32。所有节点坐标都可以显示并修改，如图 3-33。

图 3-32

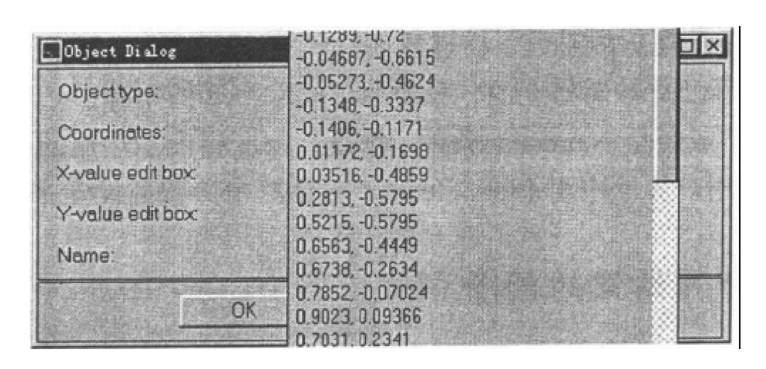


图 3-33

设定方程类型并输入边界条件。不妨设方程为 Poisson 方程，满足齐次 Dirichlet 条件，在 PDE Specification 对话框，选 Elliptic，设置 c=1，a=0、f=1。进入边界条件模式，打开 Boundary Condition 对话框、取默设设置。

作网格剖分。单击 Mesh 菜单下 Mesh Mode 选项，显示网格图，如图 3-34。再单击 Mesh 菜单下 Show Nodo Labels 选项，显示节点编号，也可以放大细看，如图 3-35、图 3-36。

如果单击 Mesh 菜单中 Export Mesh...选项，打开的 Export 对话框中有 p,e,t 三个变量，按 OK 按钮确定，可将节点坐标、边界矩阵和三角形矩阵分别输出给 p,e,t 三个变量。这时在 MATLAB 命令窗口中键入 p，按回车键，会显示出所有节点的坐标，如图 3-37。若键入 e，按回车键，则显示单元边界矩阵，键入 t 显示三角形矩阵。

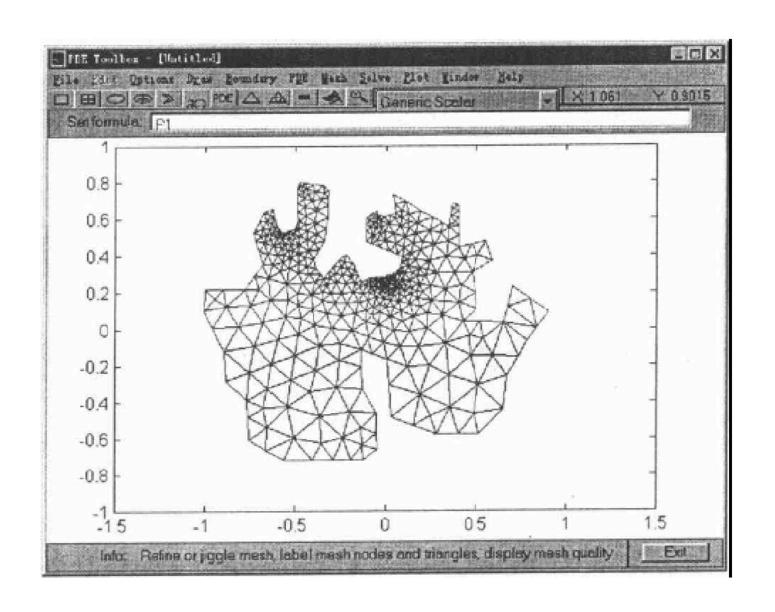


图 3-34

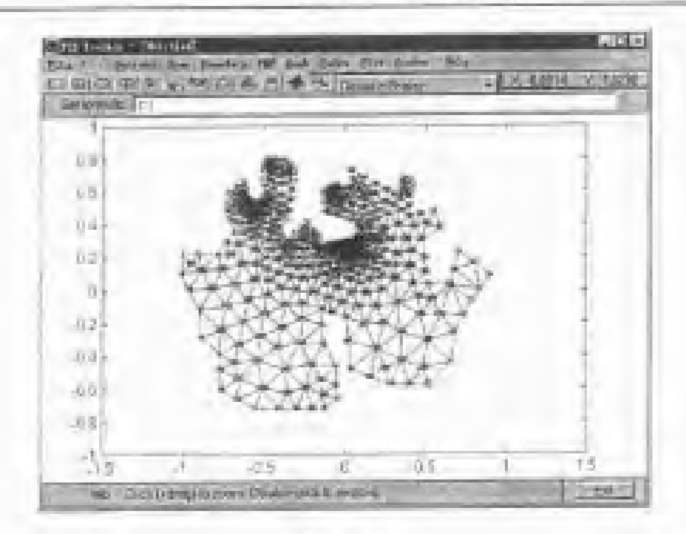


图 3-35

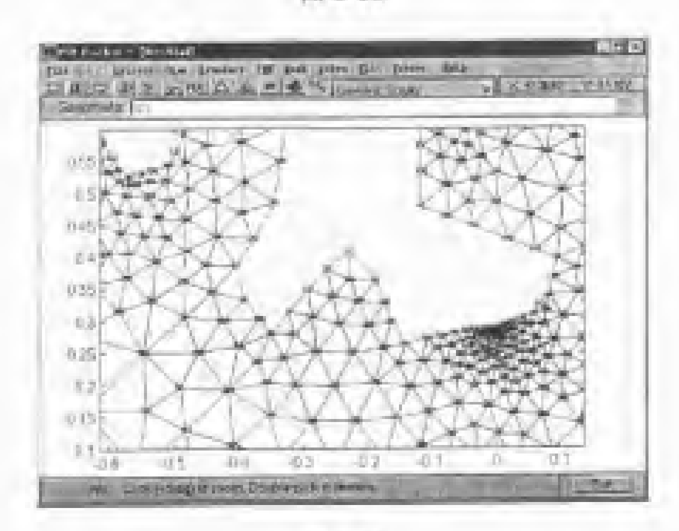


图 3-36

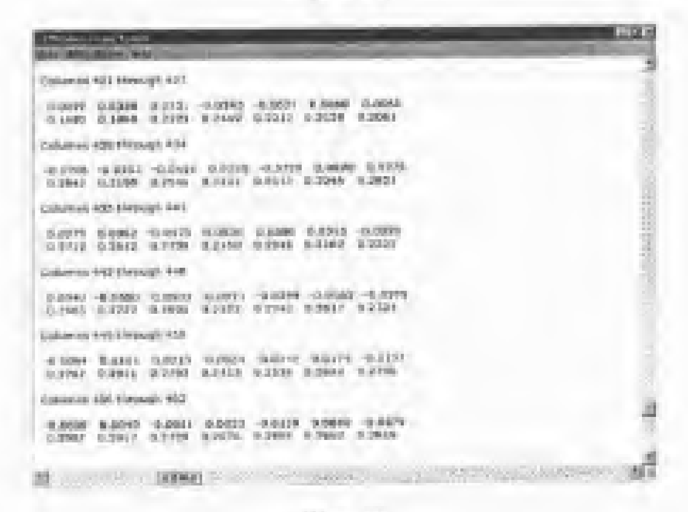


图 3-37

求解方程。执行 Solve 菜单中 Solve PDE 命令，即可求解方程。再单击 Solve 菜单中 Export Solution...选项，打开的 Export 对话框中默认显示 u 变量，单击 OK 按钮，可将解进行输出。这时在 MATLAB 的命令窗口中键入 u，按回车键，显示出按节点编号的解 u 的值，如图 3-38。



图 3-38

最后作解的图形。单击 Plot 菜单中 Parameters...选项，打开 Plot Selection 对话框，选择彩色 Color 和等值线 Contour 两项，作出解的等值线彩图，如图 3-39。若选择 Color，Height (3-D plot)及 Show mesh 三项，解的图形如图 3-40 所示。

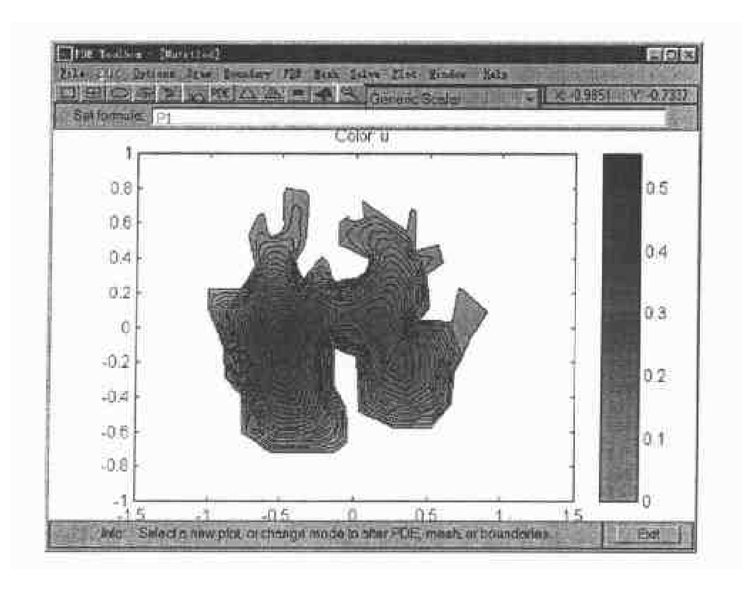


图 3-39

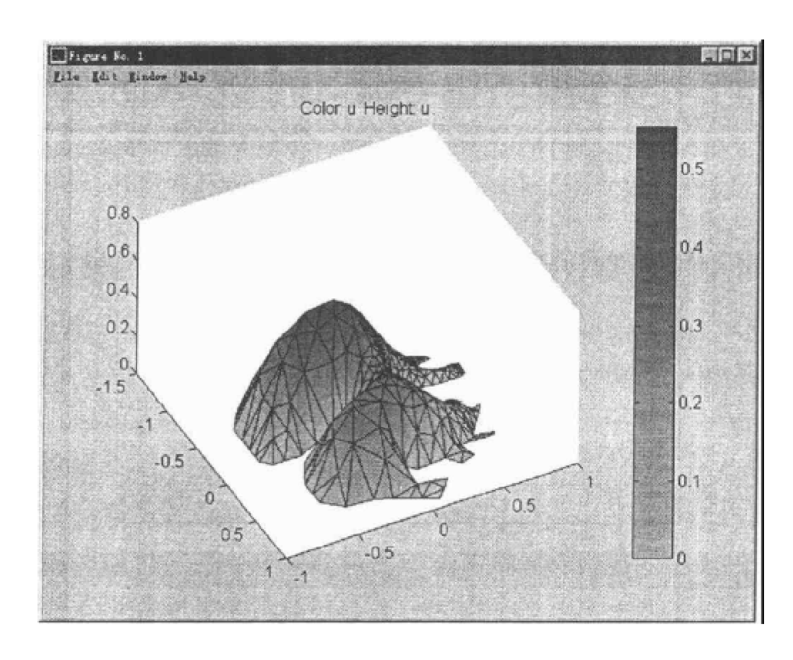


图 3-40

3.7　PDE 的 M 文件格式

在实际工程设计中，常常需要与其他有限元及绘图类软件进行交互，以取长补短。

最常用的方法是通过 DXF 文件的相互转换，而传递图形。DXF 文件是 SSAP，AUTOCAD，MICROSTATION 等软件都能接受的图形交互文件，而且它是可读可写的 ASCII 码格式，与 PDE 的 M 文件格式相同。这样我们只要将 PDE 的 M 文件中的几何描述部分读出来，再写入 DXF 文件的实体段中，即可形成 PDE 的 M 文件对 DXF 文件的接口、反过来就是 DXF 文件对 PDE 的 M 文件的接口。

下面将举一个简单的例子，描述一个平板在一点力的作用情况，并说明 PDE 的 M 文件格式。

第一步：完成几何描述部分，作多边形 P1 及一小圆 C1 (圆心为(0.6,0.6)，半径为 0.03)，如图 3-41。

第二步：进入边界模式，逐段双击边界，设置边界条件。上边界设置为 Neumann 条件，g=q=0。其余边界为齐次 Dirichlet 条件，h=1，r=0。如图 3-42。

第三步：进入 PDE 模式，设置椭圆型方程的参数。双击多边形 P1，在 PDE Specification 对话框中取 c=1.0，a=0，f=0；双击圆 C1，在 PDE Specification 对话框中取 c=1.0，a=0，f=20。

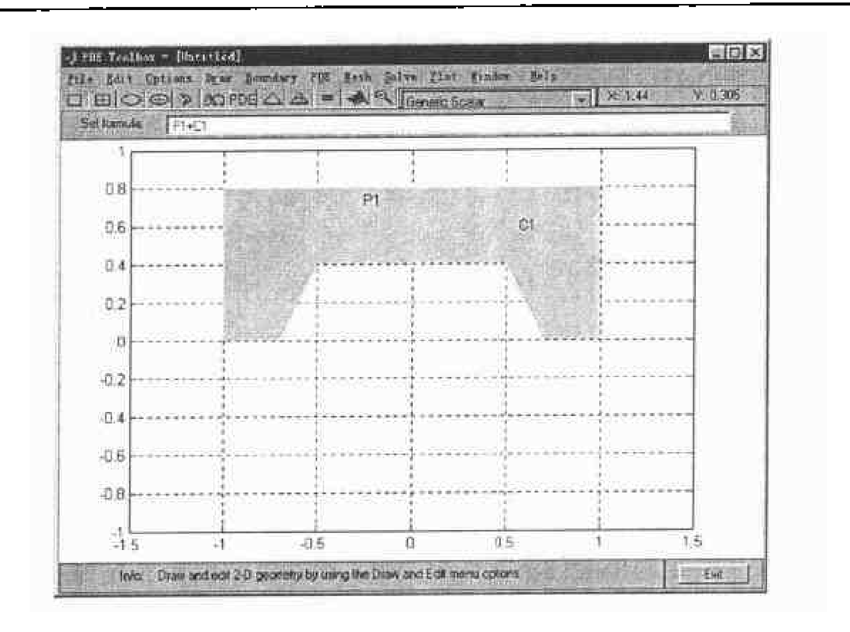


图 3-41

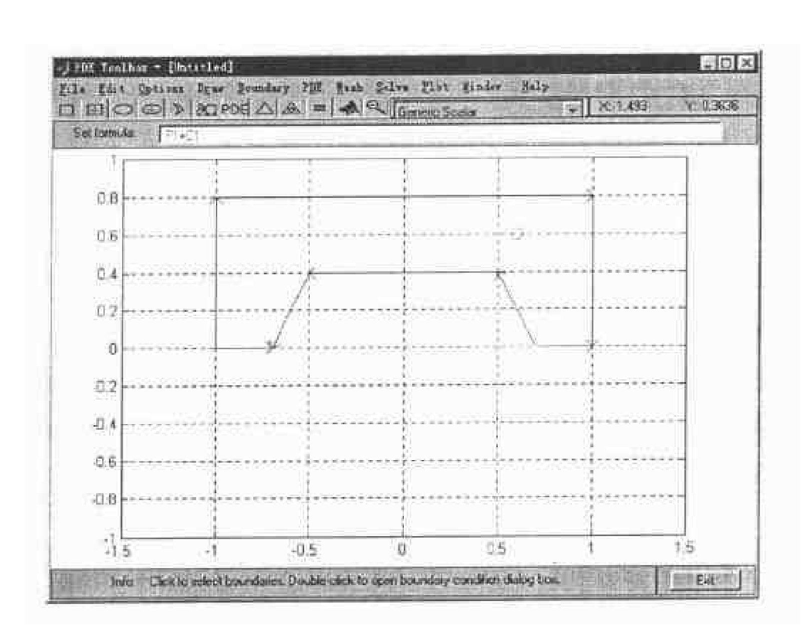


图 3-42

第四步：单击，进行网格剖分，如图 3-43。

第五步：求解。单击 Plot 菜单中 Parameters...选项，打开 Plot Selection 对话框，选择 Color，Contour，Arrows 三项，作出解的等值线、失量线彩图，如图 3-44。

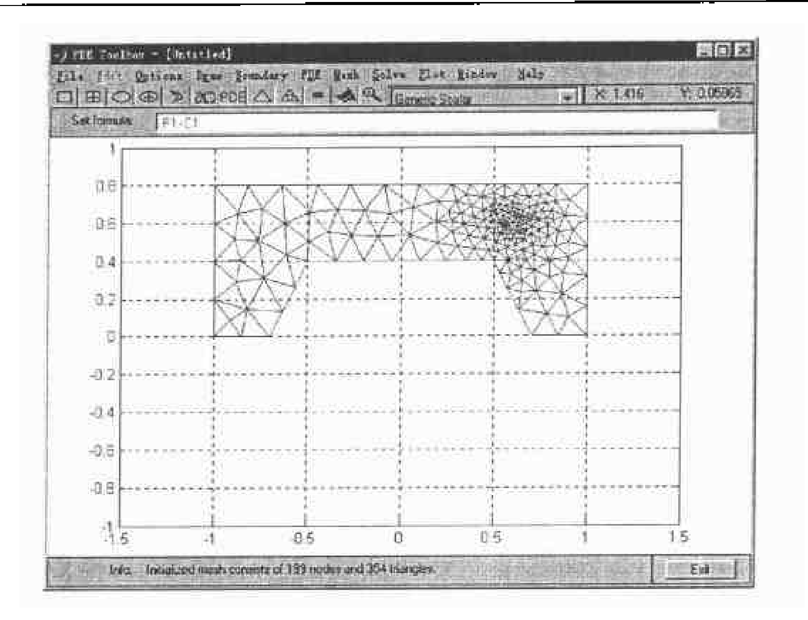


图 3-43

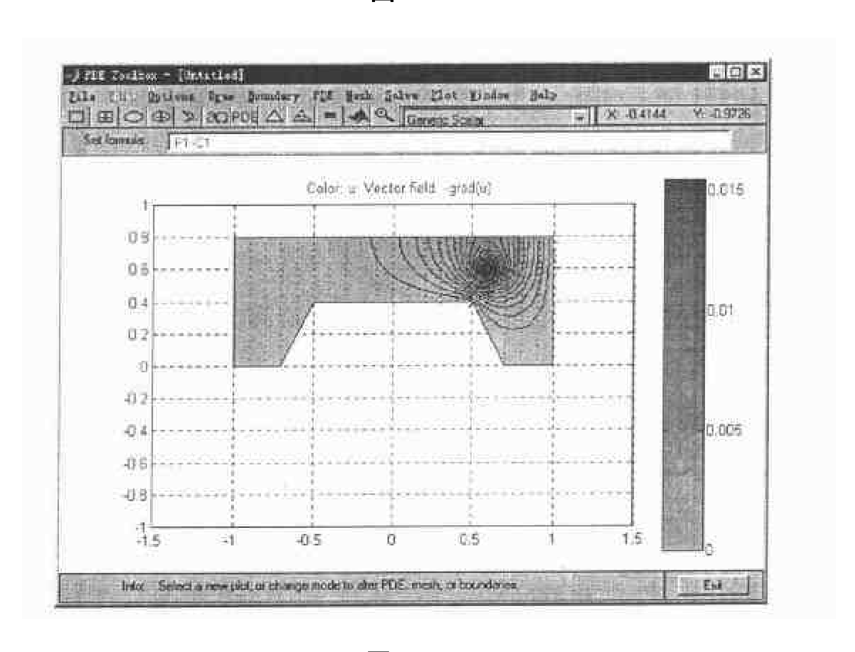


图 3-44

第六步:将 PDE Tool 图形求解过程生成 M 文件。单击 File 菜单中 Save As...命令，打开 Save As 对话框，选择文件存放的路径，给出文件名，比如键入 cad，单击保存按钮，生成一 cad.m 文件。该文件的内容如下：

```
function pdemodel
[pde_fig,ax]=pdeinit;
pdetool('appl_cb',1);
set(ax,'DataAspectRatio',[1 1 1]);
```

```
set(ax,'PlotBoxAspectRatio',[1.5 1 1]);
set(ax,'XLim',[-1.5 1.5]);
set(ax,'YLim',[-1 1]);
set(ax,'XTickMode','auto');
set(ax,'YTickMode','auto');
pdetool('gridon','on');

% Geometry description:
pdepoly([ -1,...
 1,...
 1,...
 0.7,...
 0.5,...
 -0.5,...
 -0.7,...
 -1,...
],...
[ 0.8,...
 0.8,...
 0,...
 0,...
 0.4,...
 0.4,...
 0,...
 0,...
],...
 'P1');
pdecirc(0.6,0.6,0.03,'C1');
set(findobj(get(pde_fig,'Children'),'Tag','PDEEval'),...
'String','P1+C1')

% Boundary conditions:
pdetool('changemode',0)
pdesetbd(8,...
'neu',...
1,...
'0',...
```

```
'0')
pdesetbd(7,...
'dir',...
1,...
'1',...
'0')
pdesetbd(6,...
'dir',...
1,...
'1',...
'0')
pdesetbd(5,...
'dir',...
1,...
'1',...
'0')
pdesetbd(4,...
'dir',...
1,...
'1',...
'0')
pdesetbd(3,...
'dir',...
1,...
'1',...
'0')
pdesetbd(2,...
'dir',...
1,...
'1',...
'0')
pdesetbd(1,...
'dir',...
1,...
'1',...
'0')
```

```
% Mesh generation:
setuprop(pde_fig,'Hgrad',1.3);
setuprop(pde_fig,'refinemethod','regular');
pdetool('initmesh')

% PDE coefficients:
pdesetcq(1,...
'1.0!1.0',...
'0.0!0.0',...
'0!10.0',...
'1.0!1.0',...
'0:10',...
'0.0',...
'0.0',...
'[0 100]')
setuprop(pde_fig,'currparam',...
['1.0!1.0';...
'0.0!0.0';...
'0!10.0';...
'1.0!1.0'])

% Solve parameters:
setuprop(pde_fig,'solveparam',...
str2mat('0','1000','10','pdeadworst',...
'0.5','longest','0','1E-4','','fixed','Inf'))

% Plotflags and user data strings:
setuprop(pde_fig,'plotflags',...
[1 1 1 1 1 1 1 1 0 0 0 1 1 1 0 1 0 1]);
setuprop(pde_fig,'colstring','');
setuprop(pde_fig,'arrowstring','');
setuprop(pde_fig,'deformstring','');
setuprop(pde_fig,'heightstring','');

% Solve PDE:
pdetool('solve')
```

3.8　用命令行解 PDE 的若干程序

下面给出解偏微分方程的若干程序，并给出详细的注释。

【例 3.8-1】　用有限元法求解并与精确解进行比较。

```
echo on
clc
%求单位圆上，边界条件为u=0的Poisson方程-div(grad(u))=1的解，
%并与其精确解进行比较
clc
%程序段一：定义几何区域，边界条件，微分方程
g='circleg';  %circleg是描述几何区域的文件名
b='circleb1';  %circleb1是描述边界条件的文件名
%通过下面的语句定义待解的微分方程
c=1;
a=0;
f=1;
%程序段二：网格初始化
[p,e,t]=initmesh(g,'hmax',1);
clc
%程序段三：反复加密网格，直到能满足误差要求
error=[]; er=1;
while er>0.001,
  [p,e,t]=refinemesh(g,p,e,t);
  u=assempde(b,p,e,t,c,a,f);  %求数值解
  exact=(1-p(1,:).^2-p(2,:).^2)'/4;
  er=norm(u-exact,'inf');
  error=[error er];
  fprintf('Error:% e. Number of nodes:%d\n',er,size(p,2));
end
%程序段四：画解的图形
pdesurf(p,t,u);

%程序段五：画误差图形
pdesurf(p,t,u-exact);
echo off
```

【例 3.8-2】 解 Helmholtz 方程并研究反射波。

```
echo on
clc
%程序段一：解Helmholtz方程-div(grad(u))-k^2u=0
%并研究正方形上的反射波，波源来自右边
clc
%入射波波数为60
k=60;
g='scatterg';  % scatterg描述几何区域的文件名,此区域为圆内有一方洞
b='scatterb';  % scatterb是描述边界条件的文件名,内边界满足
               % Dirichlet条件,外边界满足Neumann条件
%选择方程系数c,a,f
c=1;
a=-k^2;
f=0;
%程序段二：初始化网格和加密网格
[p,e,t]=initmesh(g);
[p,e,t]=refinemesh(g,p,e,t);
[p,e,t]=refinemesh(g,p,e,t);
%绘出网格图
pdemesh(p,e,t);axis equal
clc
%在复平面上求解
u=assempde(b,p,e,t,c,a,f);
%取复数解的实部
h=newplot;set(get(h,'Parent'),'Renderer','zbuffer')
pdeplot(p,e,t,'xydata',real(u),'zdata',real(u),...
'mesh','off');
colormap(cool)
clc

%制作反射波的动画程序
m=10;  %帧数
h=newplot;hf=get(h,'Parent');set(hf,'Renderer','zbuffer')
axis tight,set(gca,'DataAspectRatio',[1 1 1]);
axis off
M=moviein(m,hf);
```

```
maxu=max(abs(u));
for j=1:m,...
  uu=real(exp(-j*2*pi/m*sqrt(-1))*u);...
  fprintf('%d',j);...
  pdeplot(p,e,t,'xydata',uu,'colorbar','off',...
  'mesh','off'),...
  caxis([-maxu maxu]);...
  axis tight,set(gca,'DataAspectRatio',[1 1 1]);...
  axis off,...
  M(:,j)=getframe(hf);...
  if j==m,...
    fprintf('done\n');...
  end,...
end
%显示动画
movie(hf,M,50);
echo off
```

【例 3.8-3】 求最小曲面问题。

```
echo on
clc
%解方程-div(1/sqrt(1+grad|u|^2)*grad(u))=0,在边界上u=x^2

g='circleg';  %几何区域是单位圆
b='circleb2';  % circleb2是描述边界上u=x^2的文件名
c='1./sqrt(1+ux.^2+uy.^2)';
a=0;
f=0;
rtol=1e-3;  %非线性求解程序的容差
clc

%生成网格
[p,e,t]=initmesh(g);
[p,e,t]=refinemesh(g,p,e,t);

%求解
u=pdenonlin(b,p,e,t,c,a,f,'tol',rtol);
```

```
%显示图形解
pdesurf(p,t,u);
echo off
```

【例 3.8-4】 用子区域分解法解有限元问题。

```
echo on
clc
%在L型区域上，边界条件为u=0，用分解子区域法求Poisson方程
% -div(grad(u))=1的解
clc
%程序段一：定义问题
g='lshapeg';  %选定区域(lshapeg是描述L型区域的文件名)
b='lshapeb';  %选定边界条件(lshapeb是描述L型区域的边界上
              %u=0的文件名)
c=1;
a=0;
f=1;
time=[];  %用子区域求解命令Assempde要输入参数time
[p,e,t]=initmesh(g);  %初始化网格
[p,e,t]=refinemesh(g,p,e,t);
[p,e,t]=refinemesh(g,p,e,t);
clc
np=size(p,2);
%求出公共点
cp=pdesdp(p,e,t);
%分配空间
nc=length(cp);
C=zeros(nc,nc);  %定维数
FC=zeros(nc,1);
%组装区域1并更改其他区域
[i1,c1]=pdesdp(p,e,t,1);ic1=pdesubix(cp,c1);
[K,F]=assempde(b,p,e,t,c,a,f,time,1);
K1=K(i1,i1);d=symmmd(K1);i1=i1(d);
K1=chol(K1(d,d));B1=K(c1,i1);a1=B1/K1;
C(ic1,ic1)=C(ic1,ic1)+K(c1,c1)-a1*a1';
f1=F(i1);e1=K1'\f1;FC(ic1)=FC(ic1)+F(c1)-a1*e1;
```

```
%组装区域2并更改其他区域
[i2,c2]=pdesdp(p,e,t,2);ic2=pdesubix(cp,c2);
[K,F]=assempde(b,p,e,t,c,a,f,time,2);
K2=K(i2,i2);d=symmmd(K2);i2=i2(d);
K2=chol(K2(d,d));B2=K(c2,i2);a2=B2/K2;
C(ic2,ic2)=C(ic2,ic2)+K(c2,c2)-a2*a2';
f2=F(i2);e2=K2'\f2;FC(ic2)=FC(ic2)+F(c2)-a2*e2;

%组装区域3并更改其他区域
[i3,c3]=pdesdp(p,e,t,3);ic3=pdesubix(cp,c3);
[K,F]=assempde(b,p,e,t,c,a,f,time,3);
K3=K(i3,i3);d=symmmd(K3);i3=i3(d);
K3=chol(K3(d,d));B3=K(c3,i3);a3=B3/K3;
C(ic3,ic3)=C(ic3,ic3)+K(c3,c3)-a3*a3';
f3=F(i3);e3=K3'\f3;FC(ic3)=FC(ic3)+F(c3)-a3*e3;

%求解
u=zeros(np,1);
u(cp)=C\FC;  %公共点
u(i1)=K1\(e1-a1'*u(c1));  % SD1中的点
u(i2)=K2\(e2-a2'*u(c2));  % SD2中的点
u(i3)=K3\(e3-a3'*u(c3));  % SD3中的点

%绘图
pdesurf(p,t,u)
echo off
```

【例 3.8-5】 热传导问题的动画程序。

```
echo on
clc
%求解在正方形区域上非连续初始条件的、具有热源的典型热传导方程
% du/dt-div(grad(u))=1
clc

%定义问题
g='squareg';  %描述正方形的文件名squareg赋予符号变量g
b='squareb1';  % squareb1是正方形边界为1的边界条件文件名
c=1;
```

```
a=0;
f=1;
d=1;

%初始化网格
[p,e,t]=initmesh(g);
%初始条件：半径为0.4的圆内部取1，外部取0
u0=zeros(size(p,2),1);
ix=find(sqrt(p(1,:).^2+p(2,:).^2)<0.4);
u0(ix)=ones(size(ix));
clc
%在时间段[0,0.1]内取20个点求解
nframes=20;
tlist=linspace(0,0.1,nframes);
%解抛物型方程
u1=parabolic(u0,tlist,b,p,e,t,c,a,f,d);
clc
%为提高绘图速度，内插值成矩形网格
x=linspace(-1,1,31);y=x;
[unused,tn,a2,a3]=tri2grid(p,t,u0,x,y);

%制作动画
newplot;
Mv=moviein(nframes);
umax=max(max(u1));
umin=min(min(u1));
for j=1:nframes,...
  u=tri2grid(p,t,u1(:,j),tn,a2,a3);
  i=find(isnan(u));u(i)=zeros(size(i));...
  surf(x,y,u);caxis([umin umax]);colormap(cocl),...
  axis([-1 1 -1 1 0 1]);...
  Mv(:,j)=getframe;...
end

%显示动画
movie(Mv,10)
echo off
```

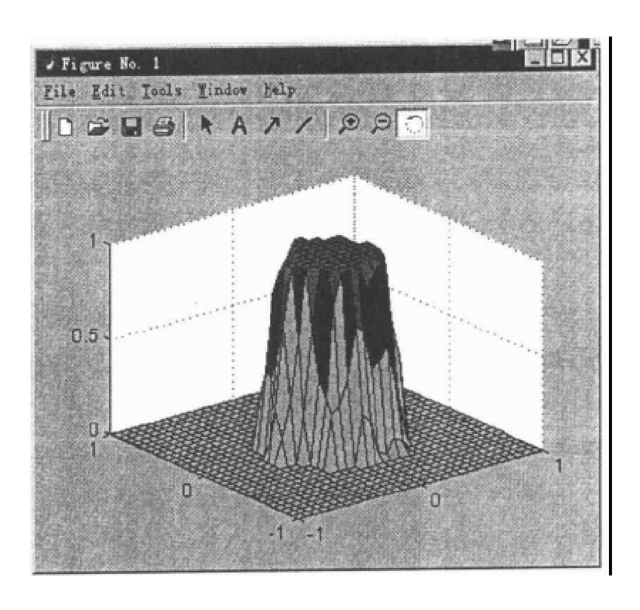


图 3-45　具有热源的典型热传导方程的图形解

【例 3.8-6】　波的传播问题的动画程序。

```
echo on
%波的传播问题的动画程序
clc

%在正方形上解标准的波动方程d2u/dt2-div(grad(u))=0
clc
%定义问题
g='squareg';   % squareg是单位正方形文件名
b='squareb3';   % squareb3是边界条件文件名，左、右边界为u=0，
                %上下边界u的法向导数为0
c=1;
a=0;
f=0;
d=1;

%初始化网格
[p,e,t]=initmesh('squareg');
clc

%初始条件:u(0)=atan(cos(pi/2*x))和
% dudt(0)=3*sin(pi*x).*exp(sin(pi/2*y))
```

```
x=p(1,:)';
y=p(2,:)';

u0=atan(cos(pi/2*x));
ut0=3*sin(pi*x).*exp(sin(pi/2*y));
clc

%在时间段[0,5]内取31个点求解
n=31;
tlist=linspace(0,5,n);

%解双曲型问题
uu=hyperbolic(u0,ut0,tlist,b,p,e,t,c,a,f,d);
clc
%为提高绘图速度、内插值成矩形网格
delta=-1:0.1:1;
[uxy,tn,a2,a3]=tri2grid(p,t,uu(:,1),delta,delta);
gp=[tn;a2;a3];

%制作动画
newplot;
M=moviein(n);
umax=max(max(uu));
umin=min(min(uu));
for i=1:n,...
  if rem(i,10)==0,...
    fprintf('%d',i);...
  end,...
  pdeplot(p,e,t,'xydata',uu(:,i),'zdata',uu(:,i),...
  'zstyle','continuous','mesh','off','xygrid',...
  'on','gridparam',gp,'colorbar','off');...
  axis([-1 1 -1 1 umin umax]); caxis([umin umax]);...
  M(:,i)=getframe;...
  if i==n,...
    fprintf('done\n');...
  end,...
end
```

```
%显示动画
nfps=5;
movie(M,10,nfps);
echo off
```

【例 3.8-7】 点源问题和自适应解。

```
echo on
clc

%在单位圆上解边界条件u=0的Poisson方程-div(grad(u))=delta(x,y),
%其精确解为u=-1/(2*pi)*log(r)，显然在原点是奇异的。用自适应方法可
%求出除原点附近以外任何点处的解。
pause  %按任意键继续
clc

%定义问题
g='circleg';  %选区域
b='circleb1';  %边界上为0
c=1;
a=0;
f='circlef';  % circlef是文件名，描述点源在原点。包含原点的三角形，
              %输出值是1/area，否则，输出值为0。

%用tripick函数输出返回errf>tol的三角形
[u,p,e,t]=adaptmesh(g,b,c,a,f,'tripick',...
'circlepick','maxt',2000,'par',1e-3);
clc

%生成自适应网格
pdemesh(p,e,t);axis equal

%画图形解
pdeplot(p,e,t,'xydata',u,'zdata',u,'mesh','off');
hold on
pdemesh(p,e,t,u);  %图形解如图3-46所示
pause  %按任意键继续
clc
```

```
%与精确解比较
x=p(1,:)';
y=p(2,:)';
r=sqrt(x.^2+y.^2);
uu=-log(r)/2/pi;
pdeplot(p,e,t,'xydata',u-uu,'zdata',u-uu,'mesh','off');
echo off
```

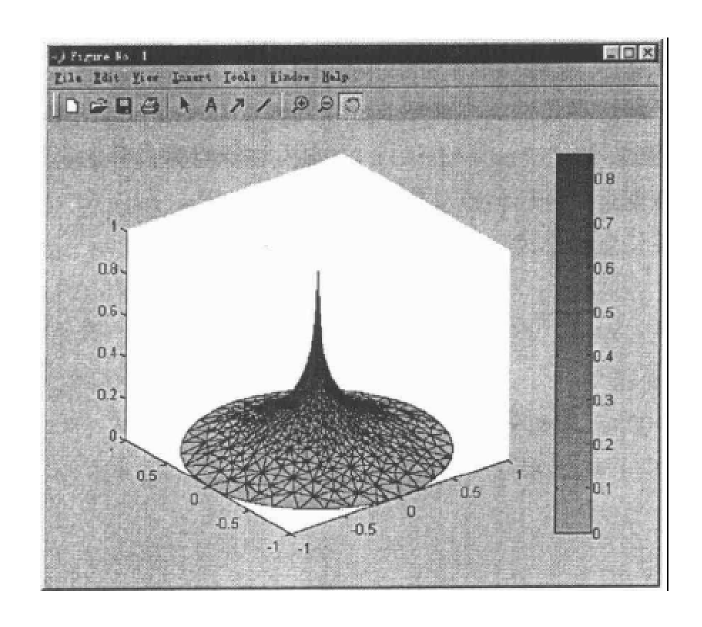


图 3-46 点源问题的图形解

【例 3.8-8】 用矩形网格求解 Poisson 方程。

```
echo on
clc
%在正方形上解带有Dirichlet边界条件的Poisson方程
% -div(grad(u))=3x^2
%将快速解与标准的PDE求解器求得的解进行比较
clc
%定义问题
g='squareg';  %取正方形
b='squareb4';  % squareb4为边界条件文件名：右端边界为半周期
               %的正弦函数，其他边界为0
c=1;
a=0;
f='3*x.^2';
```

```
%细分网格
n=16;
[p,e,t]=poimesh(g,n);
pdemesh(p,e,t); axis equal
clc
%调用快速求解器
tm=cputime;
u=poisolv(b,p,e,t,f);
cputime-tm
%用标准方法求解
tm=cputime;
u1=assempde(b,p,e,t,c,a,f);
cputime-tm
clc
%求图形解
pdesurf(p,t,u)
clc
% 65*65个节点的粗网格
n=64;
[p,e,t]=poimesh(g,n);
clc
%调用快速求解器poisolv
tm=cputime;
u=poisolv(b,p,e,t,f);
cputime-tm
%用assempde求解，与快速解对比
tm=cputime;
u1=assempde(b,p,e,t,c,a,f);
cputime-tm
clc
echo off
```

第四章　PDE Toolbox 中的命令简介

本章将介绍 PDE Toolbox 中的命令函数。利用这些命令函数，可通过命令行而不是用 PDE Tool 图形用户界面来解偏微分方程。为便于查阅，先以表格形式给出命令函数的意义，然后有选择地对其中一些命令作详细的介绍。

4.1　PDE Toolbox 中的函数及其分类

表 4-1　　PDE 数值计算函数

函　数	目　　的
adaptmesh	生成自适应网格和解 PDE 问题
assema	组装积分区域贡献
assemb	组装边界条件贡献
assempde	组装刚度矩阵和方程右边的函数
hyperbolic	解双曲型 PDE 问题
parabolic	解抛物型 PDE 问题
pdeeig	解特征值 PDE 问题
pdenonlin	解非线性 PDE 问题
poisolv	求矩形网格上泊松方程的快速解

表 4-2　用户界面算法

函　数	目　的
pdecirc	画圆
pdeellip	画椭圆
pdemdlcv	转化为 1.0 版本的 M 文件
pdepoly	画多边形
pderect	画矩形
pdetool	进入 PDE 工具箱的用户界面

表 4-3　几 何 算 法

函　数	目　的
csgchk	检查几何描述矩阵的有效性
csgdel	删除最小区域之间的分界线
decsg	分解结构矩阵成最小区域
initmesh	创建初始网格
jigglemesh	调整三角形网格内的点
pdearcl	在表达式参数和弧长之间插值
poimesh	在矩形区域上创建规则网格
refinemesh	加密三角形网格
wbound	写边界条件说明文件
wgeom	写几何说明文件

表 4-4 绘图函数

函数	目的
pdecont	绘制等值线图的快速命令
pdegplot	绘制 PDE 几何图
pdemesh	绘制 PDE 三角形网格图
pdeplot	一般 PDE 工具箱的绘图函数
pdesurf	绘制表面图的速写命令

表 4-5 通用算法

函数	目的
dst	离散正弦变换
pdeadgsc	用相对误差判别准则选择最低劣的三角形
pdeadworst	相对最差值选择低劣三角形
pdecgrad	计算 PDE 的通量
pdeent	与已知三角形单元相邻的三角形单元的索引
pdegrad	计算 PDE 解的梯度
pdeintrp	从单元节点数据到三角形单元中点的插值
pdejmps	为自适应进行误差估计
pdeprtni	从三角形中点数据到节点数据的插值
pdesde	子区域集中边缘的索引
pdesdp	子区域集中点的索引
pdesdt	子区域集中三角形的索引
pdesmesh	计算结构力学张量函数
pdetrg	三角形几何数据
pdetriq	测量三角形网格质量

续表

poiasma	泊松方程快速解的边界点矩阵的贡献
poicalc	矩形网格上泊松方程的快速解
poiindex	矩形网格正规次序的索引
sptarn	解一般稀疏矩阵特征值问题
tri2grid	由 PDE 三角形网格转换成矩形网格

表 4-6　　用户定义算法

函　数	目　的
pdebound	创建边界 M 文件
pdegeom	创建几何 M 文件

4.2 PDE 数值计算函数简介

adaptmesh

生成自适应网格并求 PDE 的解。

格式：[u,p,e,t]=adaptmesh(g,b,c,a,f)

[u,p,e,t]=adaptmesh(g,b,c,a,f,'PropertyName',PropertyValue,...)

说明　[u,p,e,t]=adaptmesh(g,b,c,a,f,'PropertyName',PropertyValue,...)产生自适应网格并求 PDE 的解。所给的选项为属性名和属性值对。调用此函数将求出定义在 Ω 上的椭圆标量方程

$$-\nabla\cdot(c\nabla u)+au=f\ ,$$

或椭圆方程组

$$-\nabla\cdot(\boldsymbol{c}\otimes\nabla \boldsymbol{u})+\boldsymbol{au}=\boldsymbol{f}$$

的解。g 和 b 是几何区域和边界条件。p,e,t 是网格数据。

解 u 用向量表示。关于解向量表达方式的详细描述参见 assempde 部分。

此函数在多次对三角形网格进行加密的基础上求 PDE 问题的解序列。通过调用没有选项的函数 initmesh 产生第一次网格；然后再把第一次生成网格作为选项作用 adaptmesh。作用 adaptmesh 时，首先是解方程，估计误差，根据误差选择三角形集合，最后将这些三角形加密，以获得第二次三角形网格；再重新计算 PDE 的解……如此循环直到不再有三角形可供选择，或已达到三角形的最大数，或已生成的三角形网格达到要生成的最大数才停止。

g 是 PDE 问题的几何区域。g 既可以是分解几何矩阵，又可以是几何 M 文件名。分解几何矩阵和几何 M 文件的格式可分别参见 decsg 和 pdebound。

b 是 PDE 问题的边界条件。b 可以是边界条件矩阵也可以是边界 M 文件名。边界条件矩阵和边界 M 文件的格式可分别查阅 assemb 和 pdebound。

PDE 问题的自适应网格由网格数据 p,e,t 给出。可通过 initmesh 查阅网格数据的表达式。PDE 问题的系数 c,a,f 可以由多种方法给出。在含有 adatmeshd 的文本中，如果用属性 Nonlin 激活非线性求解器，那么系数可以是 u 的函数。但系数不能是时间 t 的函数。所有选项的完整格式可查阅 assempde。

下面的表 4-7 列举了属性名、属性值、对应的缺省值以及属性的说明。

表 4-7

属性名	属 性 值	缺 省 值	说 明
Maxt	止整数	Inf	生成新三角形的最大个数
Ngen	正整数	10	生成三角形网格的最大次数
Mesh	p1,e1,t1	initmesh	初始网格
tripick	MATLAB 函数	pdeadworst	三角形选择方法
par	数值	0.5	函数参数
Rmethod	longestlregular	longest	三角形网格的加密方法
Nonlin	onloff	off	使用非线性求解器
Toln	数值	1E-4	非线性允许误差
Init	u0	0	非线性初始值
Jac	fixedllumpedlfull	fixed	非线性雅可比矩阵的计算
Norm	numericlinflenergy	Inf	非线性残差范数

par 用于函数 tripick 中(tripick 函数将在下面介绍)。一般它作为方程与解的符合程度的允许误差。

三角形网格的加密次数不会超过 Ngen 次。当网格中三角形的个数超过 Maxt 时，加密也将被停止。

p1,e1,t1 是输入网格数据。这个三角形网格数据作为自适应算法的初始网格数据。而这种网格数据的表达方式可查阅 initmesh。如果没有提供初始网格数据，可调用没有选项的 initmesh，将结果作为初始网格数据。

tripick 是用户自定义三角形选择方法。给出一个由函数 pdejmps 算出的误差估计值，三角形选择方法选择在下一个生成三角形中要加密的三角形。用变量 p,t,cc,aa,ff,u,errf,par 调用此函数。p,t 表示当前三角形的生成。cc,aa,ff 是展开到三角形中点处当前 PDE 问题的系数。u 是当前解。errf 是算出的误差估计值。par 是 adaptmesh 所需的选项参数。cc,aa,ff,u,errf 矩阵都有 Nt 列。Nt 是当前三角形的个数。cc,aa,ff 的行数与 c,a,f 的行数相同。对方程组的每个方程都有 errf 的一行与之对应。pdeworst 和 pdeadgsc 是 PDE 工具箱中的两个三角形标准选择方法。pdeworst 在 errf 超过最坏值的某个比值(缺省值是 0.5)的地方选择三角形；而 pdeadgsc 用相对误差选择三角形。自适应算法也可以解非线性 PDE 方程。对于非线性 PDE 问题，参数 Nonlin 必须设为 on。非线性允许误差 Toln、非线性初始值 u0、非线性雅可比矩阵 Jac 和非线性残差范数 Norm 必须通过非线性求解器 pdenonlin。关于非线性求解器可查阅 pdenonlin。

【例 4.2-1】 在扇形区域上求解拉普拉斯方程，方程在弧上满足 Dirichlet 条件 u=cos(2/3*atan2(y,x))，在直线上 u=0。并与精确解进行比较。用最坏误差法则加密三角形网格直到获得不少于 500 个三角形的网格数据为止。

```
[u,p,e,t]=adaptmesh('cirsg','cirsb',1,0,0,'maxt',...
500,'tripick','pdeadworst','ngen',inf);
x=p(1,:);y=p(2,:);
exact=((x.^2+y.^2).^(1/3).*cos(2/3*atan2(y,x)))';
max(abs(u-exact))
ans=
  0.0028
size(t,2)
ans=
  629
pdemesh(p,e,t);axis equal
```

最大绝对误差为 0.0028，具有 629 个三角形，如图 4-1。如果用大小相同的三角形网格，则需要加密多少次？

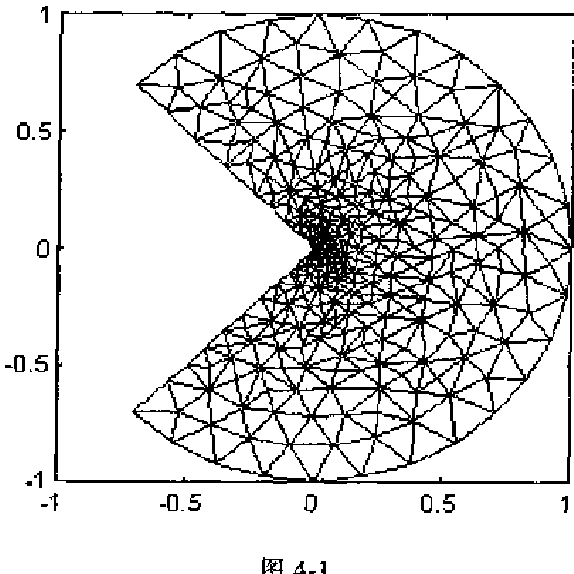


图 4-1

```
[p,e,t]=initmesh('cirsg');
[p,e,t]=refinemesh('cirsg',p,e,t);
[p,e,t]=refinemesh('cirsg',p,e,t);
u=assempde('cirsb',p,e,t,1,0,0);
x=p(1,:);y=p(2,:);
exact=((x.^2+y.^2).^(1/3).*cos(2/3*atan2(y,x)))';
max(abs(u-exact))
ans=
    0.0078
size(t,2)
ans=
    3152
```

这说明最大绝对误差仅仅只有 0.0078，而三角形网格中的三角形个数却有 3152 个。再加密一次：

```
[p,e,t]=refinemesh('cirsg',p,e,t);
u=assempde('cirsb',p,e,t,1,0,0);
x=p(1,:);y=p(2,:);
exact=((x.^2+y.^2).^(1/3).*cos(2/3*atan2(y,x)))';
max(abs(u-exact))
ans=
    0.0050
size(t,2)
```

```
ans=
   12608
    pdemesh(p,e,t);axis equal
```

也就是说，用尺寸相同的三角形加密网格，要 12608 个三角形才能得到比用自适应方法要好的绝对误差。用 pdemesh(p,e,t)语句绘出此网格图，可见已无法用肉眼辨认图中的三角形。

注意：当用相同三角形加密时，三角形数目增加四倍，误差只减小 0.6。对于有正常解的问题来说，希望误差是 $O(h^2)$。但是，由于在原点 $u \approx r^{1/3}$，所以，此问题的解是奇异的。

结束时将显示下列信息：

- Adaption completed (意思是函数 tripick 返回 0 个要加密的三角形)
- Maximum number of triangles obtaine
- Maximum number of refinement passes obtained

assema

组装积分区域贡献。

格式：[K,M,F]=assema(p,t,c,a,f)
　　　[K,M,F]=assema(p,t,c,a,f,u0)
　　　[K,M,F]=assema(p,t,c,a,f,u0,time)
　　　[K,M,F]=assema(p,t,c,a,f,u0,time,sdl)
　　　[K,M,F]=assema(p,t,c,a,f,time)
　　　[K,M,F]=assema(p,t,c,a,f,time,sdl)

说明　[K,M,F]=assema(p,t,c,a,f)组装刚度矩阵 K、质量矩阵 M 和方程右边的向量 F。

输入参数 p,t,c,a,f,u0,time 和 sdl 与 assempde 中的参数意义一样。

assemb

组装边界条件贡献。

格式：[Q,G,H,R]=assemb(b,p,e)
　　　[Q,G,H,R]=assemb(b,p,e,u0)
　　　[Q,G,H,R]=assemb(b,p,e,u0,time)
　　　[Q,G,H,R]=assemb(b,p,e,u0,time,sdl)
　　　[Q,G,H,R]=assemb(b,p,e,time)

[Q,G,H,R]=assemb(b,p,e,time,sdl)

说明 [Q,G,H,R]=assemb(b,p,e)组装矩阵 Q 和 H 以及向量 G 和 R。Q 将加到系统矩阵，包括混合边界条件贡献；G 将加到右边函数，包括一般 Neumann 边界条件、混合边界条件的贡献。方程 H*u=R 表达了 Dirichlet 边界条件。

输入参数 p,e,u0,time 和 sdl 的意义与 assempde 中的意义一样。

输入参数 b 用来描述 PDE 问题的边界条件。它既可以是边界条件矩阵又可以是边界条件的 M 文件名。边界条件 M 文件的格式可参见 pdebound，而边界条件矩阵的格式将在下面描述。

PDE 工具箱可处理下列类型边界条件：

在一般 Neumann 边界线段上，q 和 g 通过方程 $\boldsymbol{n}\cdot(c\nabla u)+qu=g$ 与法向导数值相联系。

在 Dirichlet 线段上，$hu=r$。

PDE 工具箱还能够解区域 Ω 上的偏微分方程组。假设方程组中的变量有 N 个,那么一般的边界条件是

$$\boldsymbol{hu}=\boldsymbol{r},$$

$$\boldsymbol{n}\cdot(c\otimes\nabla\boldsymbol{u})+\boldsymbol{qu}=\boldsymbol{g}+\boldsymbol{h}'\mu,$$

其中 $\boldsymbol{n}\cdot(c\otimes\nabla\boldsymbol{u})$ 是 $N\times1$ 阶矩阵，其第 i 个分量为

$$\sum_{j=1}^{N}(\cos\alpha\, c_{i,j,1,1}\frac{\partial}{\partial x}+\cos\alpha\, c_{i,j,1,2}\frac{\partial}{\partial y}+\sin\alpha\, c_{i,j,2,1}\frac{\partial}{\partial x}+\sin\alpha\, c_{i,j,2,2}\frac{\partial}{\partial y})u_j\,.$$

此处 α 是边界法向量与 x 轴之间的角，而法向量的方向指向区域 Ω 的外部。

边界条件矩阵由 pdetool 创建(确切地说，应该称为 pdetool 函数)。然后，用函数 assemb 来组装边界贡献并赋值给矩阵 Q,G,H 和 R。以后也可以把边界条件矩阵用 wbound 函数存储为边界条件 M 文件。

分解几何矩阵(Decomposed Geometry Matrix)的每一列都是边界条件矩阵的对应列。每一列的格式必须遵循下面的规则：

第 1 行是方程组的维数 N。

第 2 行是 Dirichlet 边界条件数 M。

从第 3 行到第 $3+N^2-1$ 行是用字符串表示 $\boldsymbol{q}$ 的长度。这个长度按与 $\boldsymbol{q}$ 有关的列方向的次序存储。

从第 $3+N^2$ 行到 $3+N^2+N-1$ 行包括表示 $\boldsymbol{g}$ 的字符串的长度。

从第 $3+N^2+N$ 行到 $3+N^2+N+MN-1$ 行包括表示 $\boldsymbol{h}$ 的字符串的长度。这个长度按与 $\boldsymbol{h}$ 有关的列方向的次序存储。

从第 $3+N^2+N+MN$ 行到 $3+N^2+N+MN+M-1$ 行包括表示 $\boldsymbol{r}$ 的字符串的长度。

接下来的行包括 MATLAB 文本表达式所表示的真实边界条件函数。如上所述，文本字符串也有一个长度。这个长度按与矩阵 **h** 和 **q** 有关的列方向的次序存储。两个字符串之间没有分隔符。可以插入含有下列变量的文本表达式：

• 二维坐标 x 和 y。

• 边界线段参数 s，弧长的比率。在边界线段的起始处 s=0，然后朝着箭头所指的方向逐渐增加到 1。

• 外法向量分量 nx 和 ny。如果需要切向量，可以用 tx 和 ty 来表示，其中 tx=-ny, ty=nx。

• 解 u (除非已指定输入参数 u)。

• 时间 t (除非已指定输入参数 time)。

【例 4.2-2】 下面的例子描述了边界条件矩阵的格式。对于标量(N =1) PDE 问题，其边界条件设为 Neumann 边界条件($M = 0$)：

$$\boldsymbol{n}\cdot(c\nabla u)=-x^2,$$

这个边界条件将被表示为列向量：

[1 0 1 5 ′0′ ′–x.^2′]′.

注意：此处没有存储 h 或 r 的字符串长度。

同样，对于标量 PDE 问题，其边界条件设为 Dirichlet 边界条件：

$$u=x^2-y^2,$$

它被存储为列向量：

[1 1 1 1 1 9 ′0′ ′0′ ′1′ ′x.^2-y.^2′]′.

对于带有混合边界条件($M = 1$)的方程组($N = 2$)，其边界条件设为

$$(h_{11},h_{12})\boldsymbol{u}=r_1,$$

$$\boldsymbol{n}\cdot(\boldsymbol{c}\otimes\nabla\boldsymbol{u})+\begin{pmatrix}q_{11} & q_{12}\\ q_{21} & q_{22}\end{pmatrix}\boldsymbol{u}=\begin{pmatrix}g_1\\ g_2\end{pmatrix}+s,$$

将存储为如下的列向量：

2
1
lq11
lq21
lq12
lq22
lg1
lg2

lh11
lh12
lr1
q11 ...
q21 ...
q12 ...
q22 ...
g1 ...
g2 ...
h11 ...
h12 ...
r1 ...

此处 lq11,lq21,…表示 MATLAB 文本表达式的长度，q11,q21,…表示真实表达式。

用 PDE Tool 也可创建边界条件。方法是在边界模式中双击边界以键入边界条件；然后在 Boundary 菜单中选择 Export Decomposed Geometry, Boundary Cond's...选项，将边界条件输出到 MATLAB 的主工作窗口。

assempde

组装刚度矩阵和 PDE 方程右边的函数矩阵。

格式：u=assempde(b,p,e,t,c,a,f)
u=assempde(b,p,e,t,c,a,f,u0)
u=assempde(b,p,e,t,c,a,f,u0,time)
u=assempde(b,p,e,t,c,a,f,time)
[K,F]=assempde(b,p,e,t,c,a,f)
[K,F]=assempde(b,p,e,t,c,a,f,u0)
[K,F]=assempde(b,p,e,t,c,a,f,u0,time)
[K,F]=assempde(b,p,e,t,c,a,f,u0,time,sdl)
[K,F]=assempde(b,p,e,t,c,a,f,time)
[K,F]=assempde(b,p,e,t,c,a,f,time,sdl)
[K,F,B,ud]=assempde(b,p,e,t,c,a,f)
[K,F,B,ud]=assempde(b,p,e,t,c,a,f,u0)
[K,F,B,ud]=assempde(b,p,e,t,c,a,f,u0,time)
[K,F,B,ud]=assempde(b,p,e,t,c,a,f,time)
[K,M,F,Q,G,H,R]=assempde(b,p,e,t,c,a,f)

[K,M,F,Q,G,H,R]=assempde(b,p,e,t,c,a,f,u0)

[K,M,F,Q,G,H,R]=assempde(b,p,e,t,c,a,f,u0,time)

[K,M,F,Q,G,H,R]=assempde(b,p,e,t,c,a,f,u0,time,sdl)

[K,M,F,Q,G,H,R]=assempde(b,p,e,t,c,a,f,time,sdl)

u=assrmpde(K,M,F,Q,G,H,R)

[K1,F1]=assrmpde(K,M,F,Q,G,H,R)

[K1,F1,B,ud]=assrmpde(K,M,F,Q,G,H,R)

说明 assrmpde 是 PDE 工具箱中的基本函数。它用有限元法来组装 PDE 问题。assrmpde 命令在区域Ω上组装标量方程：

$$-\nabla\cdot(c\nabla u)+au=f,$$

或组装方程组：

$$-\nabla\cdot(\boldsymbol{c}\otimes\nabla\boldsymbol{u})+a\boldsymbol{u}=\boldsymbol{f}.$$

这个命令可有选择地产生 PDE 问题的解。

对于标量方程，解向量 u 被表示为一个列向量，且与 p 所表示的节点相对应。对于有 n_p 个节点的 N 维方程组，u 的前 n_p 个值描述了 u 的第 1 个子块；接下来的 n_p 个值作为 u 的第 2 个子块等等。也就是说，u 的各个子块都被置于向量 u 中作为 N 块节点值。

u=assempde(b,p,e,t,c,a,f)根据从线性方程组中消去 Dirichlet 边界条件(约束处理)的边界点来组装和解 PDE 问题。

[K,F]=assempde(b,p,e,t,c,a,f)用刚度弹性逼近 Dirichlet 边界条件来组装 PDE 问题。K 和 F 分别是刚度矩阵和方程右边的函数矩阵。PDE 问题的 FEM (有限元法)公式解是 u=K\F。

[K,F,B,ud]=assempde(b,p,e,t,c,a,f)用从线性方程组中删去 Dirichlet 边界条件的边界点来组装 PDE 问题。在非 Dirichlet 条件点上的解为：u1=K\F，而完全的 PDE 问题的解可以由 MATLAB 矩阵 u=B*u1+ud 来得到。

[K,M,F,Q,G,H,R]=assempde(b,p,e,t,c,a,f)给出一个 PDE 问题的分解表达式。

u=assempde(K,M,F,Q,G,H,R)将分解表达式分解成单个的矩阵或向量的形式，然后从方程组中删去 Dirichlet 边界条件的边界点再解 PDE 问题。

[K1,F1]=assrmpde(K,M,F,Q,G,H,R)根据带有大的弹性系数的固定 Dirichlet 边界条件来分解表达式成单独的矩阵或向量。

[K1,F1,B,ud]=assrmpde(K,M,F,Q,G,H,R)根据从线性方程组中删去 Dirichlet 边界条件的边界点来分解表达式成单独的矩阵或向量形式。

输人变量 b 描述 PDE 问题的边界条件。它既可以是边界条件矩阵又可以是边界 M 文件。边界条件矩阵和边界 M 文件的格式可分别参见 assemb 和

pdebound。

PDE 问题的几何区域由网格数据 p,e,t 给出，关于网格数据表达方式的详细阐述可参见 initmesh。

输入变量 sdl 是子区域标识选项表。其作用是依表中所标识的子区域来限制组装过程。输入变量 u0 和 time 分别用于非线性解和时间步长算法。试验型输入解向量的格式与 u 的格式相同。

关于标量方程的系数

在标量方程中的系数 c,a,f 可用下列方法表示成 MATLAB 变量 c,a,f：

- 一个常数。
- 一个在三角形单元的质量中心处的值的列向量。
- 一个在三角形单元的质量中心处计算系数值的 MATLAB 文本表示。这个文本表示用上下文中的行向量 x,y,sd,u,ux,uy 和 t 来计算，而这些向量都是三角形单元质量中心处的行向量表达值。(t 是标量)这些行向量包括 x 坐标、y 坐标、子区域标识、解、解的关于 x 和 y 的导数以及时间。只有当 u0 通过了 assempde 才能使用 u,ux 和 uy。同样，只有当 t 作为时间通过了 assempde 才能使用 t。
- 用感叹号分开的 MATLAB 文本表达式序列。其中的每一表达式的表示法则与上面所叙述的法则一致。序列中表达式的数目必须与三角形表格 t 中子域数相等(这个数可通过键入 max(t(4,:))来检查)。
- 用户定义的，输入变量为(p,t,u,time)的 MATLAB 函数的函数名。如果相应的参数没有通过 assempde，则 u 和 time 将是一个空矩阵。p 和 t 是网格数据，u 是输入变量 u0。time 是作为时间输入于 assempde 的输入变量。如果 time 是 NaN (NaN 表示“非数”，如“0/0”或“∞/∞”)，而这个函数又是时间的函数，那么这个函数一定返回一个阶数正确的矩阵，但其中每个元素都是 NaN。

我们称上面所说的矩阵或 MATLAB 函数文件为系数矩阵或系数 M 文件。根据上面所叙述的任一条目，如果 c 带有两行数据，那么就写成元素为 $c_{1,1},c_{2,2}$ 的二阶对角矩阵：

$$\begin{pmatrix} c_{1,1} & 0 \\ 0 & c_{2,2} \end{pmatrix}$$

如果 c 有四行，则它们分别是二阶矩阵中的 $c_{1,1}$，$c_{2,1}$，$c_{1,2}$，$c_{2,2}$。

关于方程组的系数

设 N 是方程组的维数。则 c 是一个 N*2 的张量，即是一个 N 阶矩阵，f 是长度为 N 的列向量，c,a,d,f 的元素 cijkl,aij,dij,fi 按行方向存储在 MATLAB 矩阵 c,a,d, f 中。这些矩阵中的每一行元素的规则都与标量方程的规则相似。但有一点不同：在 MATLAB 文本表达式计算的点处，变量 u,ux,uy 都是有 N 行的矩阵，

每个分量都对应着一行。单位矩阵、对角矩阵、对称矩阵与相应的特殊情况相对应。对于张量 cijkl 这一点既适用于下标 i,j 也适用于下标 k,l。

F 中的行数由方程组的维数 N 决定。F 中的第 i 行表示 f 中的分量 fi。

a 中行数 na 与分量 aij 的关系如表 4-8 所示。对于对称的情况，组装了 i≥j 的元素，而所有不能形成的元素都将为零。

表 4-8

na	对称矩阵	aij	a 中的行数
1	No	aii	1
N	No	aii	i
N*(N+1)/2	Yes	aij	j*(j-1)/2+i
N^2	No	aij	N*(j-1)+i

下面的例子可看出方程中的矩阵 a 是怎样被存储在 MATLAB 矩阵 a 中的。方程组的系数矩阵 c 存储到 MATLAB 矩阵 c 中的法则由方程组的维数 N 和 c 中的行数 nc 所决定。c 中的行数 nc 对应于表 4-9 中第 1 列。

表中第 2 列确定了 c 的法则类型。确切的意思是对于一些较小的数，2≤N≤4，这法则仅由所作试验的次序所决定。对于对称矩阵，组装了 j≥i，l≥k 的情况，其他未提及的元素都是零。

表 4-9

nc	对称矩阵	cijkl	c 中的行数
1	No	ciill	1
2	No	ciill	l
3	Yes	ciikl	l+k-1
4	No	ciikl	2*l+k-1
N	No	ciill	i
2*N	No	ciill	2*i+k-2
3*N	Yes	ciikl	3*i+l+k-4
4*N	No	ciikl	4*i+2*l+k-6

续表

2*N*(2N-1)/2	Yes	ciikl	2*j^2+j+l+k–4
	Yes	cijkl	2*j^2–3*j+4*i+2*l+k–5
N^2	No	ciikl	4*N*(j–1)+4*i+2*l+k–6

下面的例子包括了将方程组的系数矩阵存储在 MATLAB 系数矩阵中的方法。

【例 4.2-3】 在 L 型薄膜上解方程$-\Delta u=1$，边界条件为 Dirichlet 齐次边界条件 $u=0$。最后画出解的图形。

```
[p,e,t]=initmesh('lshapeg','Hmax',0.2);
[p,e,t]=refinemesh('lshapeg',p,e,t);
u=assempde('lshapeb',p,e,t,1,0,1);
pdesurf(p,t,u)
hold on
pdemesh(p,e,t,u)
```

解的图形如图 4-2 所示。

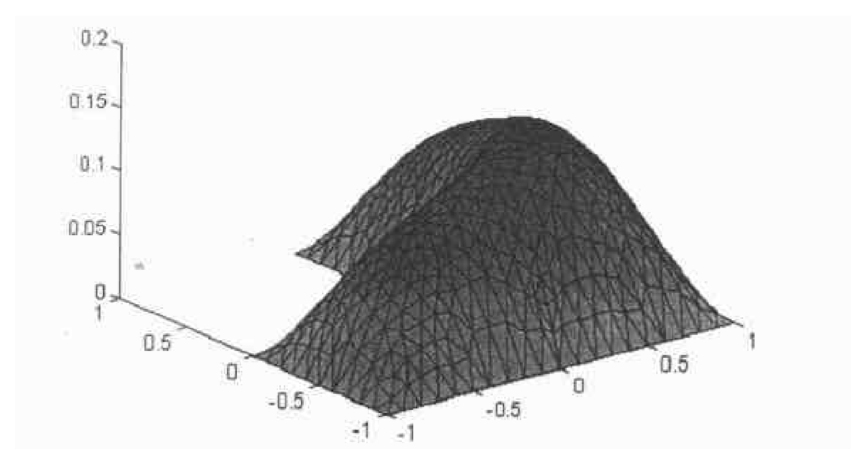


图 4-2

【例 4.2-4】 下面是三维方程组中，a 矩阵存储于 MATLAB 矩阵中的方法。

na=1:

a(1)	0	0
0	a(1)	0
0	0	a(1)

na=3:

a(1)	0	0
0	a(2)	0
0	0	a(3)

na=6:

a(1)	a(2)	a(4)
·	a(3)	a(5)
·	·	a(6)

na=9:

a(1)	a(4)	a(7)
a(2)	a(5)	a(8)
a(3)	a(6)	a(9)

符号·处表示矩阵元素是对称的。

下面是三维方程组中，张量 c 存储在 MATLAB 矩阵中的方法。

nc=1:

c(1)	0	0	0	0	0
0	c(1)	0	0	0	0
0	0	c(1)	0	0	0
0	0	0	c(1)	0	0
0	0	0	0	c(1)	0
0	0	0	0	0	c(1)

nc=2:

c(1)	0	0	0	0	0
0	c(2)	0	0	0	0
0	0	c(1)	0	0	0
0	0	0	c(2)	0	0
0	0	0	0	c(1)	0
0	0	0	0	0	c(2)

nc=3：

c(1)	c(2)	0	0	0	0
·	c(3)	0	0	0	0
0	0	c(1)	c(2)	0	0
0	0	·	c(3)	0	0
0	0	0	0	c(1)	c(2)
0	0	0	0	·	c(3)

符号 · 处表示矩阵元素是对称的。

nc=4:

c(1)	c(3)	0	0	0	0
c(2)	c(4)	0	0	0	0
0	0	c(1)	c(3)	0	0
0	0	c(2)	c(4)	0	0
0	0	0	0	c(1)	c(3)
0	0	0	0	c(2)	c(4)

nc=3 比 nc=N 的情况更重要，况且不是所有的 nc=N 情形都是有效的。

nc=6:

c(1)	0	0	0	0	0
0	c(2)	0	0	0	0
0	0	c(3)	0	0	0
0	0	0	c(4)	0	0
0	0	0	0	c(5)	0
0	0	0	0	0	c(6)

nc=9:

c(1)	c(2)	0	0	0	0
·	c(3)	0	0	0	0
0	0	c(4)	c(5)	0	0
0	0	·	c(6)	0	0
0	0	0	0	c(7)	c(8)
0	0	0	0	·	c(9)

nc=12:

c(1)	c(3)	0	0	0	0
c(2)	c(4)	0	0	0	0
0	0	c(5)	c(7)	0	0
0	0	c(6)	c(8)	0	0
0	0	0	0	c(9)	c(11)
0	0	0	0	c(10)	c(12)

nc=21:

c(1)	c(2)	c(4)	c(6)	c(11)	c(13)
·	c(3)	c(5)	c(7)	c(12)	c(14)
·	·	c(8)	c(9)	c(15)	c(17)
·	·	·	c(10)	c(16)	c(18)
·	·	·	·	c(19)	c(20)
·	·	·	·	·	c(21)

nc=36:

c(1)	c(3)	c(13)	c(15)	c(25)	c(27)
c(2)	c(4)	c(14)	c(16)	c(26)	c(28)
c(5)	c(7)	c(17)	c(19)	c(29)	c(31)
c(6)	c(8)	c(18)	c(20)	c(30)	c(32)
c(9)	c(11)	c(21)	c(23)	c(33)	c(35)
c(10)	c(12)	c(22)	c(24)	c(34)	c(36)

hyperbolic

解双曲型 PDE 问题。

格式：u1=hyperbolic(u0,ut0,tlist,b,p,e,t,c,a,f,d)

u1=hyperbolic(u0,ut0,tlist,b,p,e,t,c,a,f,d,rtol)

u1=hyperbolic(u0,ut0,tlist,b,p,e,t,c,a,f,d,rtol,atol)

u1=hyperbolic(u0,ut0,tlist,K,F,B,ud,M)

u1=hyperbolic(u0,ut0,tlist,K,F,B,ud,M,rtol)

u1=hyperbolic(u0,ut0,tlist,K,F,B,ud,M,rtol,atol)

说明 调用此命令将得到定义在Ω上的方程

$$d\frac{\partial^2 u}{\partial t^2}-\nabla\cdot(c\nabla u)+au=f$$

或方程组

$$d\frac{\partial^2 \boldsymbol{u}}{\partial t^2}-\nabla\cdot(c\otimes\nabla \boldsymbol{u})+a\boldsymbol{u}=f$$

的解。假设此方程或方程组满足初始值 u0 和初始导数 ut0；b 为此问题的边界条件；p,e,t 为网格数据。

对于标量的情形，解矩阵 u1 中的每一行对应于 p 的列所给出的坐标处的解。u1 中的每一列对应着 tlist 中的列给出的时刻的解。对于有 np 个节点的 N 维方程组来说，u1 的前 np 行是 u 的第 1 子块；第 2 个 np 行是 u 的第 2 子块；如此下去，即节点行的 N 个子块组成了 u。

输入变量 b 是边界条件。它既可以是边界条件矩阵又可以是边界 M 文件。边界条件可以依赖时间 t。边界条件矩阵和边界 M 文件的格式可分别参见 assemb 和 pdebound。

PDE 问题的几何区域由网格数据 p,e,t 给出。网格数据的详细表达方式可参见 initmesh。

PDE 问题系数可以有两种表示方式，由 c,d,a,f 给出，且它们可以是时间 t 的函数。在 assempde 中给出了所有可选择的完整列表。

控制变量 atol 和 rtol 分别是绝对误差和相对误差。

对于具有初始值 u0 和 ut0 的 ODE (常微分方程)问题：

$$B'MB\frac{\mathrm{d}^2 u_i}{\mathrm{d}t^2}+K\cdot u_i=F,\quad u=Bu_i+u_d,$$

只要键入

u1=hyperbolic(u0,ut0,tlist,K,F,B,ud,M)

就可以得解。

【例 4.2-5】 已知在$-1\leqslant x,y\leqslant 1$上的波动方程：

$$\frac{\partial^2 u}{\partial t^2}=\Delta u.$$

当$x=-1$，$x=1$时，$u=0$。

当$y=-1$，$y=1$时，$\frac{\partial u}{\partial n}=0$。

初始条件为：$u(0)=\arctan(\cos\frac{\pi}{2}x)$，$\frac{\mathrm{d}u(0)}{\mathrm{d}t}=3\sin\pi x\exp(\cos\pi y)$。

求此问题在 t=0,1/6,1/3,…,29/6,5 时刻的值。

可键入：

```
[p,e,t]=initmesh('squareg');
x=p(1,:)';
y=p(2,:)';
u0=atan(cos(pi/2*x));
ut0=3*sin(pi*x).*exp(cos(pi*y));
tlist=linspace(0,5,31);
uu=hyperbolic(u0,ut0,tlist,'squareb3',p,e,t,1,0,0,1);
```

parabolic

解抛物型PDE问题。

格式：u1=parabolic(u0,tlist,b,p,e,t,c,a,f,d)

u1=parabolic(u0,tlist,b,p,e,t,c,a,f,d,rtol)

u1=parabolic(u0,tlist,b,p,e,t,c,a,f,d,rtol,atol)

u1=parabolic(u0,tlist,K,F,B,ud,M)

u1=parabolic(u0,tlist,K,F,B,ud,M,rtol)

u1=parabolic(u0,tlist,K,F,B,ud,M,rtol,atol)

说明　u1=parabolic(u0,tlist,g,b,p,e,t,c,a,f,d)用有限元法解在区域 Ω 上具有网格数据p,e,t并带有边界条件b和初始值u0的偏微分方程

$$d\frac{\partial u}{\partial t}-\nabla\cdot(c\nabla u)+au=f$$

或偏微分方程组

$$d\frac{\partial \boldsymbol{u}}{\partial t}-\nabla\cdot(\boldsymbol{c}\otimes\nabla \boldsymbol{u})+\boldsymbol{au}=\boldsymbol{f}.$$

对于标量方程，解矩阵u1中的每一行都是p中对应列给出的坐标处的解。u1中的每一列都是tlist中所对应的条目给出的时刻的解。对于有np个节点，维数是N的方程组，u1的前np行表示u的第1个分量；u1的第2个np行表示u的第2个分量等等。也就是说，u的分量在向量u中作为节点行的N块。输入变量b是PDE问题的边界条件。b既可以是边界条件矩阵也可以是边界M文件。边界条件可以依赖于时间t。在assemb和pdebound中分别描述了边界条件矩阵和边界M文件的格式。

PDE问题的几何区域由网格数据p,e,t给出。关于网格数据的描述可在initmesh中找到。

PDE问题的系数c,a,d,f 可以用多种方法表示，它们也可以是时间t的函数。

所有选项的完整表格都可以在assempde中找到。

atol 和 rtol 分别是通过了 PDE 问题求解器的绝对误差和相对误差。

u1=parabolic(u0,tlist,K,F,B,ud,M)求带有初始条件 u0 的 ODE 问题

$$B'MB\frac{\mathrm{d}u_i}{\mathrm{d}t}+K\cdot u_i=F,\quad u=Bu_i+u_d$$

的解。

【例 4.2-6】 在几何区域$-1\leqslant x,y\leqslant 1$ (squareg)上，当$x^2+y^2<0.4^2$时，$u(0)=1$，其他区域上$u(0)=0$，且满足 Dirichlet 边界条件$u=0$ (squarebl)，求在时刻(0,0.1,20)处热传导方程$\frac{\partial u}{\partial t}=\Delta u$的解。

```
[p,e,t]=initmesh('squareg');
[p,e,t]=refinemesh('squareg',p,e,t);
u0=zeros(size(p,2),1);
ix=find(sqrt(p(1,:).^2+p(2,:).^2)<0.4);
u0(ix)=ones(size(ix));
tlist=linspace(0,0.1,20);
u1=parabolic(u0,tlist,'squareb1',p,e,t,1,0,1,1);
```

pdeeig

解特征值 PDE 问题。

格式：[v,l]=pdeeig(b,p,e,t,c,a,d,r)

[v,l]=pdeeig(K,B,M,r)

说明 [v,l]=pdeeig(b,p,e,t,c,a,d,r)用有限元法产生定义在Ω上 PDE 特征值问题的解。

标量PDE特征值问题：$-\Delta\cdot(c\nabla u)+au=\lambda du$.

PDE组特征值问题：$-\nabla\cdot(\boldsymbol{c}\otimes\nabla\boldsymbol{u})+\boldsymbol{au}=\lambda\boldsymbol{du}$.

p,e,t 描述几何区域；b 描述边界条件；r 是两个元素的向量，标出实数轴上的一个区间(这个区间的左端点可以是-inf)。调用此命令后，返回此区间上的所有特征值并存于 l 中。

v 是特征向量矩阵。对于标量 PDE 特征值问题，v 的每一列都是 p 所对应的节点处的解值的特征向量。对于带有 np 个节点，维数是 N 的 PDE 组特征值问题，v 的第 1 个 np 行描述 v 的第 1 个分量；v 的第 2 个 np 行描述 v 的第 2 个分量等等。也就是说，v 的分量作为节点行的 N 块存放于 v 中。

b 描述了 PDE 问题的边界条件。b 既可以是边界条件矩阵，也可以是边界 M 文件。边界条件矩阵和边界 M 文件的格式分别在 assemb 和 pdebound 中描述。

注意：特征值问题是一个齐次问题，即边界条件中g=0, r=0。非齐次部分被自动地删去。

PDE问题几何区域由网格数据p,e,t给出。关于网格数据的表达方式在initmesh中给出。

PDE问题的系数c,a,d可以用各种方法给出。在pdeeig的上下文中系数不能依赖u，也不能依赖时间t。所有选项完整表格能在assempde中找到。

[v,l]=pdeeig(K,B,M,r)产生一般稀疏矩阵特征值问题的解。此问题为：$Ku_i=\lambda B'MBu_i$，$u=Bu_i$，其中λ的实部在r的区间中。

【例4.2-7】 在L型区域上对于$-\Delta u=\lambda u$计算小于100的特征值和所对应的特征模态。并显示第1和第16个特征模态。

```
[p,e,t]=initmesh('lshapeg');
[p,e,t]=refinemesh('lshapeg',p,e,t);
[p,e,t]=refinemesh('lshapeg',p,e,t);
[v,l]=pdeeig('lshapeb',p,e,t,1,0,1,[-Inf 100]);
l(1)  %取第1个特征值
pdesurf(p,t,v(:,1))  %绘制第1个特征模态，如图4-3
figure
membrane(1,20,9,9)  % MATLAB函数
figure
l(16)  %取第16个特征值
pdesurf(p,t,v(:,16))  %绘制第16个特征模态，如图4-4
```

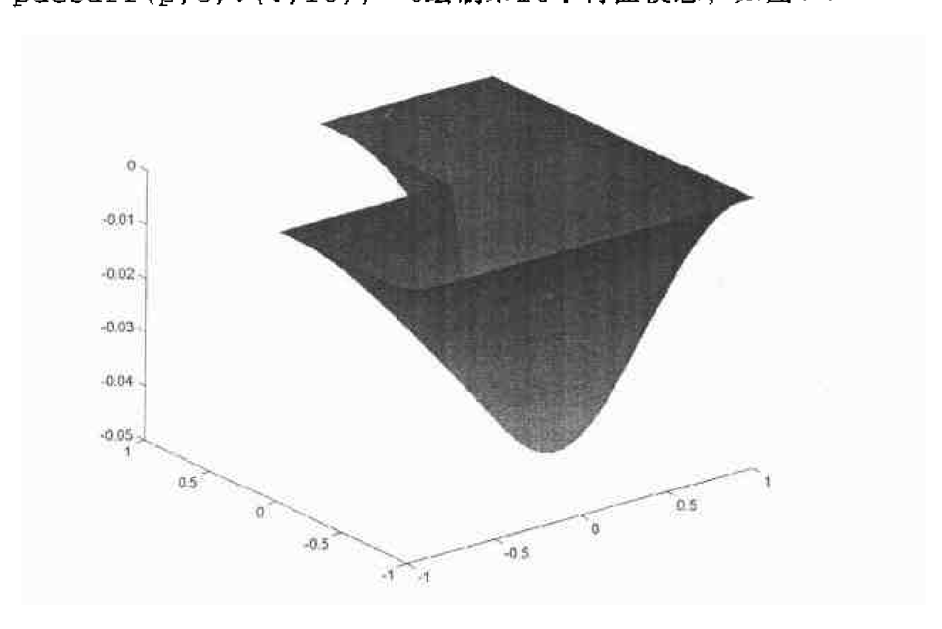


图 4-3 第一特征模态图

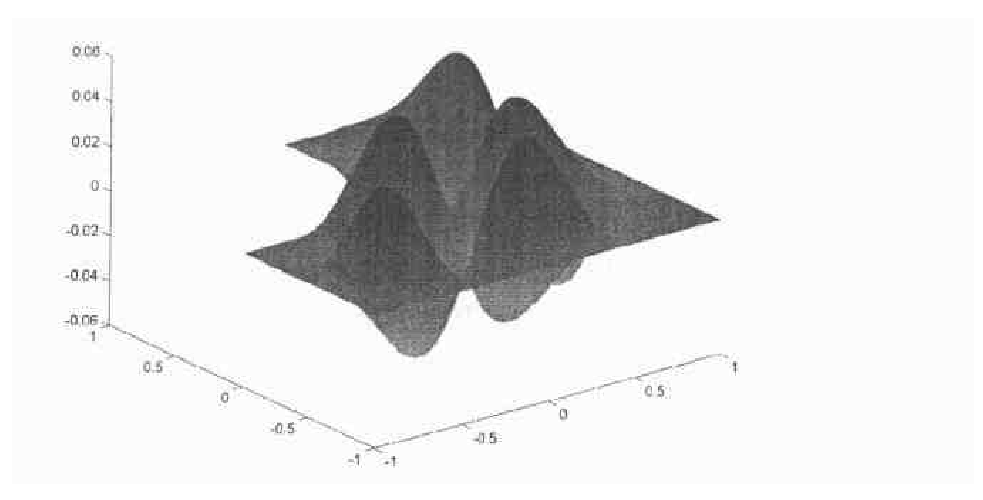


图 4-4 第 16 特征模态图

注意：在标准情况下，全部的区域内部中c和d都是正数。所有的特征值都是正的，此时最好将区间的左端点选为零。下面对c=0或d=0两种情形作如下的讨论：

• 如果在一个子区域内 d=0，则质量矩阵 M 为奇异矩阵。如果处处 c>0，那么这一点不会引出任何麻烦。矩阵对(K,M)将有一个无穷多特征值的集合。

• 如果在一个子区域内 c=0，则刚度矩阵 K 为奇异矩阵。此时矩阵对(K,M)有许多零特征值。如果是一个包括零的区间，那么势必要花许多时间去寻找所有的零特征值。当然最好是选择一个既能把零排除在外又能把最小的正特征值包括在内的正数作为区间的左端点。

• 如果存在一个区域，其中 c=0, d=0 均成立，那么就得到一个奇异矩阵对。整个特征值问题是不确定的，任何值都可作为特征值。

还有一些较为棘手的情形可通过 pdeeig 来检测。如果位移矩阵是奇异的，则可试图检查另外的位移矩阵。如果这新被考察的位移矩阵仍然是奇异的，那么较合理的猜想是整个矩阵对(K,M)都是奇异的。

如果遇到的不是标准情形，那么只有利用一般物理问题的知识去解释计算结果。

pdenonlin

解非线性 PDE 问题。

格式：[u,res]=pdenonlin(b,p,e,t,c,a,f)

[u,res]=pdenonlin(b,p,e,t,c,a,f,'PropertyName',PropertyValue,...)

说明 [u,res]=pdenonlin(b,p,e,t,c,a,f)用于求解定义在 Ω 上的非线性标量 PDE 问题：

$$-\nabla\cdot(c\nabla u)+au=f\ ,$$

或非线性方程组 PDE 问题：

$$-\nabla\cdot(c\otimes\nabla \boldsymbol{u})+\boldsymbol{a}\boldsymbol{u}=\boldsymbol{f}\ .$$

此处系数 c,a,f 可以是 u 的函数。通过用具有 Armijo-Goldstein 线性搜索策略的阻尼牛顿迭代法作为算法来求解方程。

解 u 被表达为向量 u。有关解向量的详细表达方式可参见 assempde。res 是牛顿步进残差范数。

PDE 问题的三角形网格由网格数据 p,e,t 给出。关于网格数据表达式的详细情形参见 initmesh。

b 是 PDE 问题的边界条件。b 既可以是边界条件矩阵，又可以是边界 M 文件。关于边界条件矩阵和边界 M 文件的格式可分别查阅 assemb 和 pdebound。

注意：一般调用 pdebound 时，边界条件可以是 u 的函数，且用不动点迭代策略去解非线性边界条件。

PDE 问题的系数可用多种方法给出。在含有 pdenonlin 的文本中，系数可以是 u 的函数，但不能是 t 的函数。所有选项格式的完整表格可查阅 assempde。

可通过设置下列所介绍的选项来对求解器作精细的调整：

表 4-10

属性名	属性值	缺省值	说　　明
Jacobian	fixed|lumped|full	fixed	雅可比逼近
u0	字符串或数字	0	估计的初始解
Tol	正数	1E-4	残差值
MaxIter	正整数	25	高斯-牛顿迭代的最大次数
MinStep	正数	1/2^16	搜索方向的最小阻尼
Report	on|off	off	是否输出收敛信息
Norm	字符串或数字	Inf	范数残差

当前，计算雅可比矩阵有效的方法有三种：

- 函数 numjac 的稀疏满雅可比矩阵的数值估计。
- 有限元法中的系数数值微分的“块(umped)”逼近。
- 由刚度矩阵迭代成雅可比矩阵的不动点迭代。

通过设置 Jacobian 的属性 full,lumped 或 fixed 来选择上述方法。所选方法越精细，所耗机时就越多。

可以用一个表达式给出 u0 的最初猜测。一般是一个标量或一个向量。缺

省值是 0，但这对诸如带有 Dirichlet 边界条件 $u=e^{x+y}$、形式为 $\nabla\cdot(\frac{1}{u}\nabla u)=0$ 的这类方程，就无效了。

Tol 用来控制从高斯-牛顿迭代中退出。也就是说，当残差范数小于 Tol 时，就终止迭代。通过 Norm 来选择残差中的范数。这样，MATLAB 向量范数或能量范数的 energy 都能被语法所允许。

MaxIter 和 MinStep 防止陷入无穷迭代之中。每次迭代，它们都被调用。将 eport 设置为 on 则可输出收敛信息。

诊断：如果牛顿迭代不收敛，那么将显示 “Too many iteration” 或 “Stepsize too small”。如果初始猜测所产生的矩阵包括 NaN 或 Inf 元素，那么将显示 “Unsuitable initial guess u0 (default:u0=0)”。

poisolv

求在矩形网格上泊松方程的快速解。

格式：u=poisolv(b,p,e,t,f)

说明 u=poisolv(b,p,e,t,f)在正则矩形网格上解带有 Dirichlet 边界条件的泊松方程。正弦变换和三角形解相结合的方法被应用于增加表现性。

边界条件 b 必须在所有的边界点上都满足 Dirichlet 边界条件。

网格数据 p,e,t 必须是正则矩形网格，关于网格数据的表达方式可参见 initmesh。

f 给出泊松方程的右边的函数。

关于舍入误差部分，结果与 u=assempde(b,p,e,t,1,0,f)中的舍入误差部分一样。

4.3 用户界面算法函数简介

pdecirc

画圆，修改几何图形矩阵。

格式：pdecirc(xc,yc,radius,label)

说明 画一个以(xc,yc)为圆心、radius 为半径的圆。label 为标识符，可任选，也可缺省。

pdeellip

画椭圆，修改几何图形矩阵。

格式：pdeellip(xc,yc,radiusx,radiusy,angle,label)

说明 画一个以(xc,yc)为中心，以 radiusx 为 x 半轴、radiuy 为 y 半轴的椭

圆。angle 表示半轴与坐标轴之间的夹角，label 为标识符，可任选，也可缺省。

pdepoly

画多边形，修改几何图形矩阵。

格式：pdepoly(x,y,label)

说明 以向量 x,y 所对应的元素作为顶点画多边形。label 为标识符，可任选，也可缺省。

pderect

画矩形，修改几何图形矩阵。

格式：pderect(xmin,xmax,ymin,ymax,label)

说明 以(xmin,ymin)，(xmax,ymin)，(xmax,ymax)，(xmin,ymax)为顶点画矩形。label 为标识符，可任选，也可缺省。

pdetool

进入偏微分方程工具用户界面(PDE Tool GUI)。

格式：pdetool

说明 进入 PDE Tool GUI 不需要其他参数，只需在主窗口中键入 pdetool 即可。

使用 PDE Tool GUI 可以画二维区域，定义边界条件，选择偏微分方程类型、创建和加密三角形网格，计算和显示解等等。用 PDE Tool GUI 可以使解偏微分方程非常简单和直观。

如何使用 PDE Tool GUI，请参阅本书的有关章节。

4.4 几何算法函数简介

decsg

分解结构几何成最小区域。

格式：dl=decsg(gd)

dl=decsg(gd,sf,ns)

[dl,bt]=decsg(gd)

[dl,bt]=decsg(gd,sf,ns)

[dl,bt,dl1,bt1,msb]=decsg(gd)

[dl,bt,dl1,bt1,msb]=decsg(gd,sf,ns)

说明 此函数分析所画的结构几何图形模型(CSG model)，构造由区域边界和各最小区域边界所围的不相交最小区域集合；根据 CSG model 中的对象有选择地估算一组公式。常把最小区域称为“Decomposed geometry”。它能使 PDE Toolbox 中的其他函数“理解”指定的几何区域。为了绘图的需要还要构造带有边界的第二最小区域集合。

PDE Tool 用户界面为许多目的而调用 decsg。每次要画或改变一个新的对象，为了能正确地画此对象和最小区域，PDE Tool 都要调用一次 decsg。Dlaunay 三角形算法、initmesh 函数，也需要 decsg 的输出去产生初始网格。

dl=decsg(gd,sf,ns)分裂 CSG 模型，gd 为分解几何区域 dl。CSG 模型由几何描述矩阵表示；分解几何区域由分解几何矩阵表示。decsg 返回最小几何区域，而最小几何区域是用一组公式 sf 来描述的几何区域。空间名称矩阵 ns 是文本矩阵，它把 gd 中的列与 sf 中的变量名相对应。

dl=decsg(gd) 返回所有最小区域。如同把 sf 与 gd 中所有对象的并对应一样。

另外，[dl,bt]=decsg(gd)和[dl,bt]=decsg(gd,sf,ns)返回一个布尔表格。把最初的几何对象与最小区域相对应。bt 中的列对应于 gd 中相同索引的列。bt 中的行对应于最小区域索引。

[dl,bt,dl1,bt1,msb]=decsg(gd)和[dl,bt,dl1,bt1,msb]=decsg(gd,sf,ns)返回一个与布尔表格 bt1 对应的第二最小区域 dl1。所有第二最小区域有一个连通边界。这些第二最小区域可以用 MATLAB 局部对象来绘制。这些第二最小区域可以不用最初的对象引人的边界。当两个或更多的对象没有相交的边界时，将发生这种情况。

另外，调用此命令还返回一个对每个第二最小区域的绘图命令序列 msb。第一行包括围成最小区域的边界线段数。另外的行包括由边界构造的分解几何矩阵的边界线段序列。如果边界线段标识索引比边界线段总数大，则将从得到的边界线段标识数的内容中减少边界线段的总数，且绘图方向与分解几何矩阵所给方向相反。

几何描述矩阵

几何描述矩阵 gd 描述了用 PDE Tool 画的 CSG 模型。在 PDE Tool 中从 Draw 菜单选 Export Geometry Description, Set Formula, Labels...选项，可以把当前几何描述矩阵储成变量输送到 MATLAB 的主工作窗口。

几何描述矩阵的每一列对应 CSG 模型中的一个对象。支持 4 种类型的对象。第 1 行为指定对象的类形：

- 对于圆，第 1 行是 1，第 2、第 3 行分别是圆心的 x,y 坐标，第 4 行是圆

的半径。

• 对于多边形，第 1 行是 2，第 2 行是多边形边界的线段数 n，接下来的 n 行是线段起始点的 x 坐标，再下来的 n 行是线段起始点的 y 坐标。

• 对于矩形，第 1 行是 3，其他格式与多边形的格式完全一样。

• 对于椭圆，第 1 行是 4，第 2、第 3 行分别是中心的 x,y 坐标。第 4、第 5 行是椭圆的半轴，第 6 行储存椭圆的转角。

sf 中一组公式由 ns 中变量来表示。运算符 "+"，"*"、"-" 分别表示并、交和差。算子 "+"，"*" 是同级运算，"-" 是高级运算。运算次序可用圆括号来控制。

空间名称矩阵

空间名称矩阵 ns 将 gd 中的列与 sf 中的变量名相对应。ns 中的每一列都是一空间特征序列。每个这样的特征列都将分配一个名给 gd 中相应的几何对象。这样可以在公式集 sf 中指定 gd 中的指定对象。

分解几何矩阵

分解几何矩阵 dl 包括一个用 decsg 算法构造的不相交的最小分解几何区域的表达式。每条最小区域边界线段都对应于 dl 中的一列。指定最小区域之间的分界线段为 border segments，并且指定区域外边界线段为 boundary segments。第 2 行和第 3 行的每个元素表示起点和终点的 x 坐标。第 4、第 5 行表示 y 坐标。第 6 行和第 7 行是极小区域左边或右边的标识。方向从起点到终点，如果是圆或椭圆则按顺时针方向计算。在极小区域中可能有 3 种类型的边缘线段：

• 对于圆边界线段，第 1 行是 1。第 8、第 9 行是圆心坐标。第 10 行是半径。

• 对于直线线段，第 1 行是 2。

• 对于椭圆边界线段，第 1 行是 4。第 8、第 9 行是椭圆中心坐标。第 10、第 11 行是半轴。椭圆的转角存放于第 12 行。

【例 4.4-1】 下面一组命令的功能是：启动 PDE Tool GUI 并画单位圆和边长为 1 的正方形。

```
pdecirc(C,0,1)
pderect([0 1 0 1])
```

然后在 PDE Tool 用户界面窗口的 Set formula 公式栏中键入 C1-SQ1。单击 Draw 菜单中 Export Geometry Description，Set Formula，Labels…选项，将分解几何矩阵、公式集、空间名称矩阵输出到 MATLAB 的主工作窗口。再在主窗口键入：

```
[dl,bt]=decsg(gd,sf,ns);
dl
```

```
dl =
      2.0000    2.0000     1.0000     1.0000     1.0000
           0         0    -1.0000     0.0000     0.0000
      1.0000         0     0.0000     1.0000    -1.0000
           0    1.0000    -0.0000    -1.0000     1.0000
           0         0    -1.0000          0    -0.0000
           0         0     1.0000     1.0000     1.0000
      1.0000    1.0000          0          0          0
           0         0          0          0          0
           0         0          0          0          0
           0         0     1.0000     1.0000     1.0000

    bt
bt =
           1         0
```

注意：此例是带有 5 条边缘线段的一个最小区域，其中 3 条为圆形边缘线段，2 条为直线段。

initmesh

创建初始网格数据。

格式：[p,e,t]=initmesh(g)

[p,e,t]=initmesh(g,'PropertyName',PropertyValue,...)

说明 [p,e,t]=initmesh(g)用标准几何函数 g 返回一个三角形网格数据。它用的是一个 Delaunay 三角形算法。网格尺寸根据几何形状而定。

g 描述 PDE 问题的几何形状。g 既可以是分解几何矩阵(Decomposed Geometry Matrix)又可以是几何 M 文件。关于分解几何矩阵和几何 M 文件的格式可分别参见 pdecsg 和 pdegeom 中的叙述。

输出矩阵 p,e,t 是网格数据。

在节点矩阵 p 中，第 1 行和第 2 行分别是网格节点的 x 坐标和 y 坐标。

在边界矩阵 e 中，第 1 行和第 2 行是起点和终点的索引；第 3 行和第 4 行是起点和终点的参数值；第 5 行是边界线段的顺序数；第 6 行和第 7 行分别是子区域左边和右边的标识。

在三角形矩阵 t 中，前三行按逆时针方向给出三角形顶点的次序，最后一行给出子区域的标识。

下面给出属性名和属性值的表格：

表 4-11

属 性 名	属 性 值	缺省值	说　明
Hmax	数值	估计值	边界尺寸的最大值
Hgrad	数值	1.3	网格增长比率
Box	on|off	off	保护边界框
Init	on|off	off	三角形边界
Jiggle	off|mean|min	mean	调用 jigglemesh
JiggleIter	数值	10	最大迭代次数

Hmax 属性控制网格三角形尺寸。由 initmesh 创建的三角形网格的三条边，其值不会超过 Hmax。

Hgrad 属性依最小几何区域决定网格的增长比率。缺省值是 1.3，即以 30% 的比率增长。Hgrad 取值在 1 和 2 之间。

Box 和 Init 属性与网格算法的工作方式有关。如果 Box 属性置于 on，那么就可以对在边界框以内的网格生成算法的工作方式获得一个好的理解。如果 Init 属性置于 on，就能看到边界最初的三角形剖分。用下面一组命令，可决定点 x,y 的子区域标识。

```
[p,e,t]=initmesh(dl,'hmax',inf,'init','on');
[uxy,tn,a2,a3]=tri2grid(p,t,zeros(size(p,2)),x,y);
n=t(4,tn);
```

如果这个点在这个区域的外面，那么 tn 将是 NaN，而命令 n=t(4,tn)将返回失败信息。

Jiggle 属性用于控制是否要对网格进行调整(详细情况可参阅 jiggmesh 中的解释)，可一直调整到三角形质量的中间值或最小值。JiggleIter 用来建立迭代次数的上限。

算法（initmesh 执行 Delaunay 三角形剖分算法）

(1) 在边界上设置节点。

(2) 在边界框中圈定几何区域。

(3) 三角形剖分边界。

(4) 逐条检查所考虑的(边界)三角形剖分。

(5) 在大的三角形的中点插入节点。

(6) 如果还没达到 Hmax 值，重复步骤(4)。

(7) 移动边界框。

【例 4.4-2】 在 PDE Tool 窗口建立一个简单的 L 型三角形网格。

在 MATLAB 命令窗口键入 pdetool，调入 PDE Tool 图形用户界面。首先绘制 L 型区域：选中 Options 菜单中 Grid (显示栅格)和 Snap (对齐栅格)命令，使用多边形绘图工具，很容易画出 L 型区域。再单击 Mesh 菜单中 Parameters...选项，打开 Mesh Parameters 对话框，在 Maximum edge size 栏中设置 Inf。还可以在 Mesh 菜单中选 Show Node Labels 和 Show Triangle Labels，显示节点和三角形网格标识。然后单击▲按钮，创建初始网格(也可以在 Mesh 菜单中选 Initialize Mesh 命令)，屏幕上将出现如图 4-5 所示的图形。

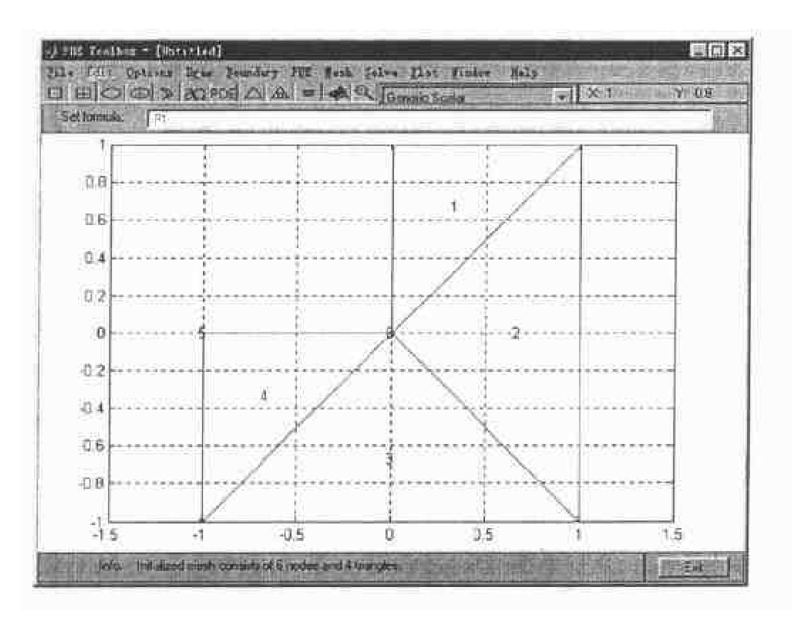


图 4-5 节点和单元编号图

通过在 Mesh 菜单中选 Export Mesh...命令，可以把相应的网格数据结构输出到主工作窗口。这时在主工作窗口分别键入 p,e,t，会有如下数框显示：

```
    p
p=
    0    1    1   -1   -1    0
    1    1   -1   -1    0    0

    e
e=
    1    2    3    4    5    6
    2    3    4    5    6    1
```

```
0    0    0    0    0    0
1    1    1    .1   1    1
1    2    3    4    5    6
0    0    0    0    0    0
1    1    1    1    1    1

t
t=
2    3    4    5
1    2    3    4
6    6    6    6
1    1    1    1
```

jigglemesh

优化微调三角形网格内部的点。

格式：p1=jigglemesh(p,e,t)

p1=jigglemesh(p,e,t,'PropertyName',PropertyValue,...)

说明　调用 p1=jigglemesh(p,e,t)的作用是通过调整节点位置来优化微调三角形网格，以提高网格质量。

下表是属性名和属性值的允许配对：

表 4-12

属性名	属性值	缺省值	说　明
Opt	off|mean|min	mean	优化方法
Iter	数值	1 或 20	最大迭代次数

每个不在边缘线上的点都将被移至由相邻三角形所形成的多边形的质量中心。这个过程根据变量 Opt 和 Iter 的设置被反复进行。

• 当 Opt 设为 off 时，这个过程被执行 Iter 次（缺省值为 1）。

• 当 Opt 设为 mean 时，反复执行此过程直到三角形的平均质量不再有效增加或已达到边界迭代次数(缺省值为 20)时为止。

• 当 Opt 设为 min 时，反复执行此过程直到最小三角形质量不再有效增加或已达到边界迭代次数(缺省值为 20)时为止。

【例 4.4-3】　创建 L 型区域的三角形网格。第 1 次不进行优化微调；第 2

次进行优化微调。

```
[p,e,t]=initmesh('lshapeg','jiggle','off');
q=pdetriq(p,t);
pdeplot(p,e,t,'xydata',q,'colorbar','on',...
'xystyle','flat')  %画未经优化的初始网格数据图
p1=jigglemesh(p,e,t,'opt','mean','iter',inf);
q=pdetriq(p1,t);
pdeplot(p1,e,t,'xydata',q,'colorbar','on',...
'xystyle','flat')  %画优化后的网格数据图
```

由图 4-6、图 4-7 可见经优化后的网格质量高得多。(在 MATLAB 图形窗口中，蓝色三角形越多，网格质量越差；红色三角形越多，网格质量越优。)

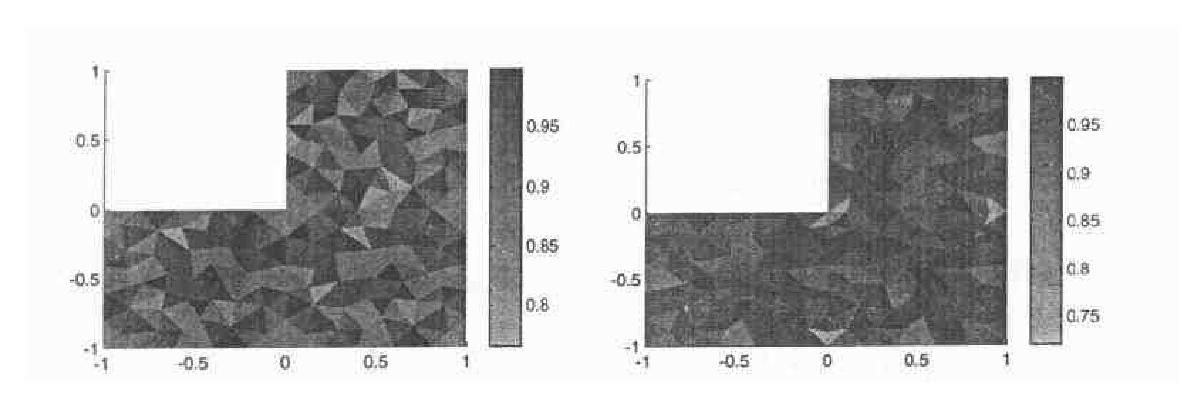


图 4-6 未经优化的三角形网格图　　图 4-7 优化后的三角形网格图

refinemesh

加密三角形网格。

格式：[p1,e1,t1]=refinemesh(g,p,e,t)

[p1,e1,t1]=refinemesh(g,p,e,t,'regular')

[p1,e1,t1]=refinemesh(g,p,e,t,'longest')

[p1,e1,t1]=refinemesh(g,p,e,t,it)

[p1,e1,t1]=refinemesh(g,p,e,t,it,'regular')

[p1,e1,t1]=refinemesh(g,p,e,t,it,'longest')

[p1,e1,t1,u1]=refinemesh(g,p,e,t,u)

[p1,e1,t1,u1]=refinemesh(g,p,e,t,u,'regular')

[p1,e1,t1,u1]=refinemesh(g,p,e,t,u,'longest')

[p1,e1,t1,u1]=refinemesh(g,p,e,t,u,it)

[p1,e1,t1,u1]=refinemesh(g,p,e,t,u,it,'regular')

[p1,e1,t1,u1]=refinemesh(g,p,e,t,u,it,'longest')

说明　[p1,e1,t1]=refinemesh(g,p,e,t)返回一个被几何区域 g、点矩阵 p、边缘矩阵 e 和三角形矩阵 t 指定的经过加密优化的三角形网格数据。

三角形网格由网格数据 p,e,t 给出。关于网格数据的表达式在 initmesh 中有详细叙述。

[p1,e1,t1,u1]=refinemesh(g,p,e,t,u)不仅加密网格，而且还用线性插值的方法将 u 扩展到新的网格上。u 的行数与 p 的列数对应，u1 的行数与 p1 元素一样多。u 的每一列分别被进行内插值。

如果输入变量 it 是一个行向量，那么它被解释为要加密的子区域的表；如果 it 是列向量，则它就是一个要加密的三角形表格。

加密的缺省方法是规则加密法：所有指定的三角形单元都被分为 4 个形状相同的三角形单元。

也能通过输入参数 longest 使用最长边加密法：把指定的每个三角形单元的最长边两等分。如果用 regular 作为最后的参数就会得到规则加密法产生的结果。用这种方法，有些被指定集合以外的三角形单元也能被加密，以保护三角形单元及其质量。

【例 4.4-4】　多次加密 L 型区域的网格，并画出每次加密后的网格图。如图 4-8 所示。

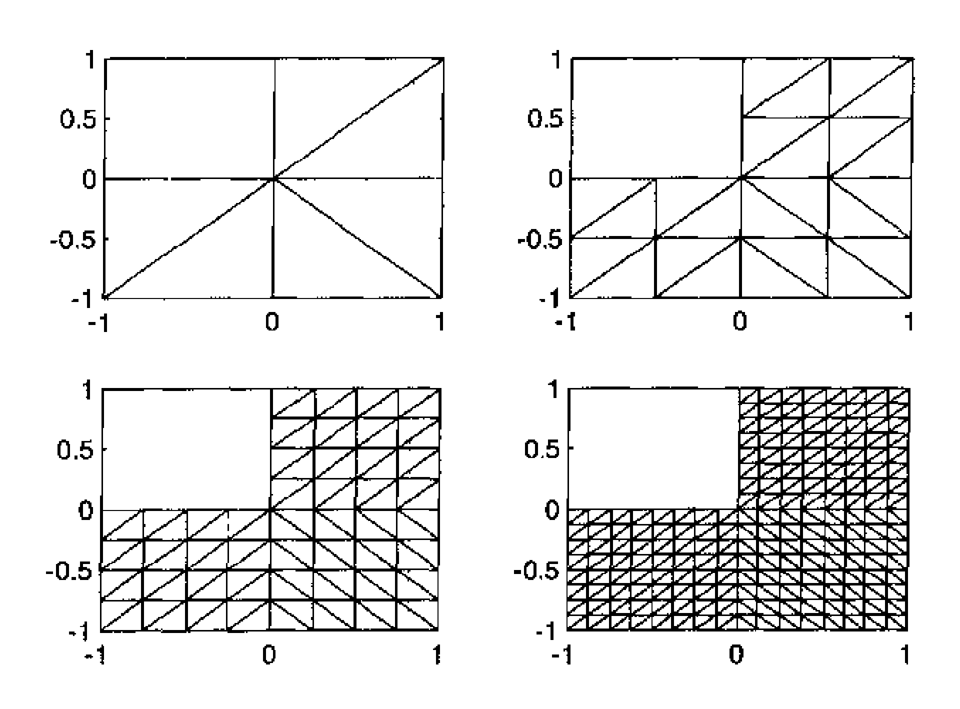


图 4-8　三角形网格加密

```
[p,e,t]=initmesh('lshapeg','hmax',inf);
subplot(2,2,1),pdemesh(p,e,t)
[p,e,t]=refinemesh('lshapeg',p,e,t);
subplot(2,2,2),pdemesh(p,e,t)
[p,e,t]=refinemesh('lshapeg',p,e,t);
subplot(2,2,3),pdemesh(p,e,t)
[p,e,t]=refinemesh('lshapeg',p,e,t);
subplot(2,2,4),pdemesh(p,e,t)
subplot
```

wbound

写边界条件的说明文件。

格式：fid=wbound(bl,mn)

说明 此命令将产生一个名为 mn.m 的边界条件说明文件。在用命令而不是用 GUI 解方程时，既可以输入说明边界条件的 M 文件，也可以输入边界条件矩阵 bl。如果因格式错误而不能写文件，将返回-1。

关于边界条件矩阵 bl 的定义参见命令 assemb。有关边界条件的 M 文件的解释，参见命令 pdebound。

wgeom

写几何区域的说明文件。

格式：fid=wgeom(dl,mn)

说明 此命令将产生一个名为 mn.m 的几何区域文件。这个 M 文件与分解几何矩阵(Decomposed Geometry Matrix) bl 等价。如果因格式错误而不能写文件，将返回-1。

关于分解几何矩阵 dl 的格式参见条目 decsg。关于几何 M 文件的格式参见条目 pdegeom。

4.5 几何绘图函数简介

pdecont

绘制等值线图的快速命令。

格式：pdecont(p,t,u)
pdecont(p,t,u,n)
pdecont(p,t,u,v)

h=pdecont(p,t,u)
h=pdecont(p,t,u,n)
h=pdecont(p,t,u,v)

说明　pdecont(p,t,u)画 10 条 PDE 问题关于节点或三角形数据解的等值线。h=pdecont(p,t,u)还返回一个关于轴的句柄值。

如果 u 是一个列向量，则组装了节点数据；如果 u 是一个列向量，则组装了三角形数据。用函数 pdeprtni 可以将三角形数据转化为节点数据。

PDE 问题的几何区域由网格数据 p 和 t 给出。有关网格数据的表达方式，可在 initmesh 中找到。

pdecont(p,t,u,v)用由 v 指定的水平线画等值线。

这个命令是下列调用的快速命令：

```
pdeplot(p,[],t,'xydata',u,'xystyle','off',...
'contour','on','levels',n,'colorbar','off');
```

如果要用更多的控制变量画等值线，就得用pdeplot代替pdecont。

【例 4.5-1】　画定义在几何区域为 L 形薄膜上的方程$-\Delta u=1$的解的等值线；边界条件为 Dirichlet 边界条件 $u=0$。

```
[p,e,t]=initmesh('lshapeg');
[p,e,t]=refinemesh('lshapeg',p,e,t);
u=assempde('lshapeb',p,e,t,1,0,1);
pdecont(p,t,u)
```

等值线如图4-9所示。

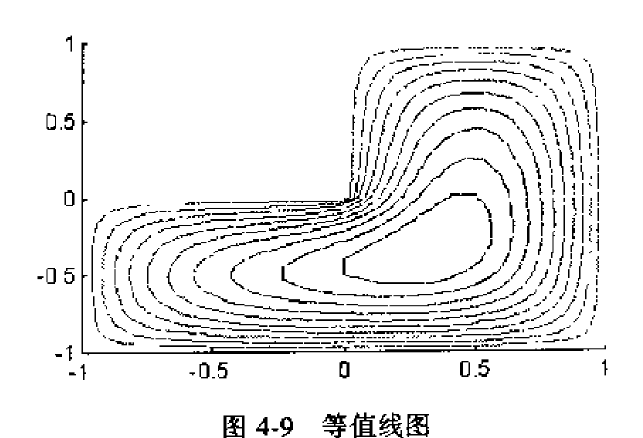


图 4-9　等值线图

pdemesh

绘制三角形网格图

格式：pdemesh(p,e,t)
　　　pdemesh(p,e,t,u)

h=pdemesh(p,e,t)

h=pdemesh(p,e,t,u)

说明 pdemesh(p,e,t)绘制由网格数据 p,e,t 指定的网格图。

h=pdemesh(p,e,t)还返回一个轴对象句柄。

pdemesh(p,e,t,u)用网格图绘制节点或三角形数据 u。如果 u 是列向量，那么组装节点数据；如果 u 是行向量，则组装三角形数据。事实上，这个绘图命令要比命令 desurf 绘图绘得快。

PDE 问题的几何区域由网格数据 p,e,t 给出。关于网格数据的详细说明可在 initmesh 中找到。

这个命令可看作是下列命令的快速作用：

pdeplot(p,e,t),

pdeplot(p,e,t,'zdata',u)

如果有更多的控制网格图形的变量，就用 pdeplot 代替 pdemesh。

【例 4.5-2】 画几何区域为 L 形薄膜的网格区域。

```
[p,e,t]=initmesh('lshapeg');
pdemesh(p,e,t)
```

平面网格如图4-10所示。

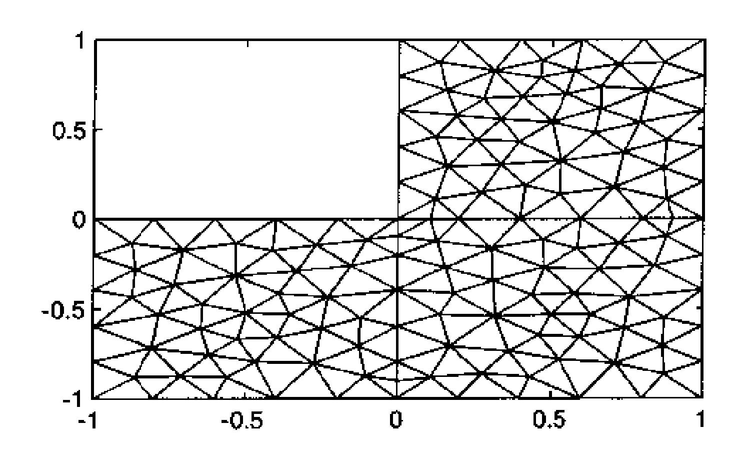


图 4-10 平面网格图

下面解定义在 L 形薄膜上的泊松方程 $-\Delta u=1$，其边界条件为齐次 Dirichlet 边界 $u=0$，并且绘出解的曲面图。

```
u=assempde('lshapeb',p,e,t,1,0,1);
pdemesh(p,e,t,u)
```

曲面网格如图 4-11 所示。

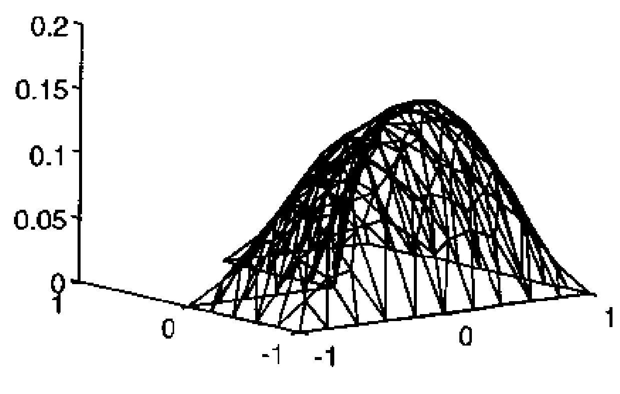


图 4-11 曲面网格图

pdeplot

是 PDE 工具箱的一般绘图函数。

格式：pdeplot(p,e,t,'PropertyName',PropertyValue,...)

h=pdeplot(p,e,t,'PropertyName',PropertyValue,...)

说明 pdeplot(p,e,t,p1,v1,...)是 PDE 工具箱的一般绘图函数。它能同时显示多个 PDE 方程的解。

PDE 问题的几何区域由网格数据 p,e,t 给出。关于网格数据表达方式的详细情形可查阅 initmesh。

正确的属性名-属性值对包括：

表 4-13

属性名	属性值{缺省}	说　　明
xydata	{data}	三角形网格数据
xystyle	off|flat|{interp}	x-y 数据绘图风格
contour	{off}|on	显示轮廓
zdata	{data}	节点或三角形数据
zstyle	off|{continuous}|discontinuous	有高度的三维绘图风格
flowdata	{data}	节点或三角形数据
flowstyle	off|{arrow}	流动图的绘图风格
colormap	{colormap} cool	x-y 彩色图名或彩色图矩阵

续表

属性名	属性值{缺省}	说　　明
xygrid	{off}lon	绘图前转换成 x-y 网格
gridparam	{[tn;a2;a3]}	三角形索引和改写调用 tri2grid 的参数
mesh	{off}lon	在图中显示网格
colorbar	offl{on}	显示彩色图柄
title	{''}	绘制文本标题
levels	{10}	水平线数或指定向量的水平线数

pdeplot 既可在 PDE Tool GUI 内部使用，又可在命令行中使用。它能同时显示 3 个对象。

可以由表面图来实现 xydata 的可视化。平面的或内插值(缺省)阴影图能被用于表面图的绘制。如果将 contour 设置为 on，那么就可以在曲面图上加上黑白的轮廓或用彩色单独画轮廓。通过显示高度来实现 zdata 的可视化。通过内插法(缺省)，三角形既可画为立体的又可画为平面的。可以像 MATLAB 箭头图那样，通过画箭头来使流动数据可视化。所有数据类型既可以是节点数据又可以是三角形数据(流动数据只能是三角形数据)。节点数据由长度是 size(p,2)的列向量表示，而三角形数据由长度是 size(t,2)的行向量表示。如果没有 xydata,zdata 或 flowdata 支持，则 pdeplot 画由 p,e,t 指定的网格。

选项 mesh 显示或隐藏(缺省)图中的网格。选项 xygrid 首先转换数据成 x-y 数据(用 tri2grid)，然后再调用标准的 MATLAB 绘图算法。属性 gridparam 使 tri2grid 数据通过 pdeplot，并快速制成动画。属性 colormap 用任意的 MATLAB 色彩图或色彩矩阵表现图。colorbar 给图增加一个颜色句柄。title 在图中插入一个标题。levels 仅用来增加一个轮廓图；给一个整数，就画一条依此数值为水平线的等值线，给一个向量，就画此向量各分量所示的等值线。

另外，h=pdeplot(p,t,u)返回一个画轴对象的句柄。

【例 4.5-3】 下面的命令是画定义在 L 形区域上的泊松方程的解的三维图形。

```
[p,e,t]=initmesh('lshapeg');
u=assempde('lshapeb',p,e,t,1,0,1);
pdeplot(p,e,t,'xydata',u,'zdata',u,'mesh','off')
```

解的三维图形如图 4-12 所示。

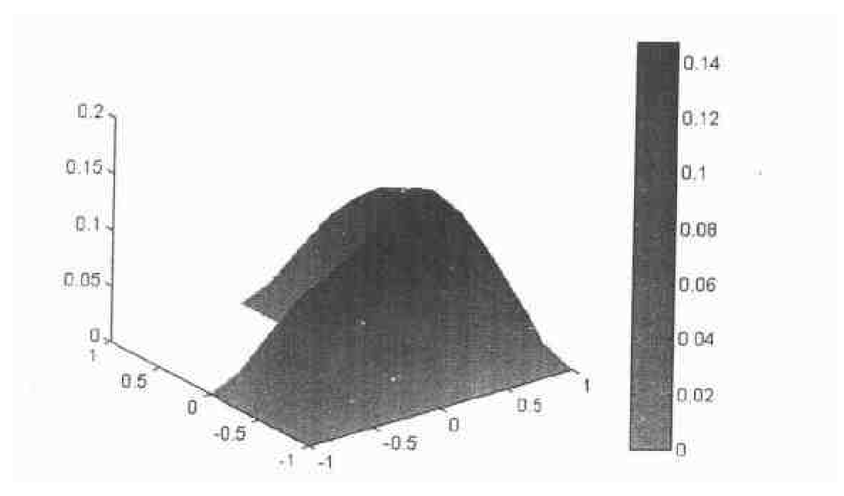


图 4-12

4.6 通用算法

pdeadgsc

用相对误差判别准则选择三角形。

格式：bt=pdeadgsc(p,t,c,a,f,u,errf,tol)

说明 bt 中元素是加密了的三角形的索引。PDE 问题的几何区域由网格数据 p 和 t 给出。

关于 p 和 t 的更详细说明可参阅 initmesh。

c, a 和 f 是 PDE 问题的系数。详细情况可参阅 assempde。

列向量 u 是当前解。也可参阅 assempde。

errf 是由 pdejmps 算出的误差指标。

tol 是容差参数。

用准则 errf>tol*scale 选择三角形；此处对 scale 作如下计算：分别设 cmax, amax, fmax, umax 为 c, a, f, u 的量大值，设 l 是包含区域的矩形的最小边。则 scale=max(fmax*l^2,amax*umax*l^2,cmax*umax)。此式使容差参数独立于方程和几何区域的缩放比例。

pdeadworst

选择最低劣值的三角形。

格式：bt=pdeadworst(p,t,c,a,f,u,errf,wlevel)

说明 bt=pdeadworst(p,t,c,a,f,u,errf,wlevel)的返回量 bt 是被加密过的三角形的索引。这些三角形将被作进一步的加密。

PDE 问题的几何区域由网格数据 p 和 t 给出。关于 p 和 t 的详细介绍可参阅 initmesh。

c, a 和 f 是 PDE 问题的系数，详细情况可参阅 assempde。

列向量 u 是当前解，详细介绍可参阅 assempde。

errf 是由 pdejmps 算出的误差指标。

wlevel 是相对于最低劣误差的误差水平，取 0 到 1 之间的值。

用 errf>wlevel*max(errf)原则选择三角形。

pdecgrad

计算 PDE 的通量。

格式：[cgxu,cgyu]=pdecgrad(p,t,c,u)

[cgxu,cgyu]=pdecgrad(p,t,c,u,time)

[cgxu,cgyu]=pdecgrad(p,t,c,u,time,sdl)

说明 [cgxu,cgyu]=pdecgrad(p,t,c,u)输出在每个三角形中心处的通量,即 $c\times\Delta u$。

cgxu 的第 i 行是

$$\sum_{j=1}^{N} c_{ij11}\frac{\partial u_j}{\partial x}+c_{ij12}\frac{\partial u_j}{\partial y}.$$

cgyu 的第 i 行是

$$\sum_{j=1}^{N} c_{ij21}\frac{\partial u_j}{\partial x}+c_{ij22}\frac{\partial u_j}{\partial y}.$$

对于 t 中的每一个三角形，cgxu 和 cgyu 中都有一列与其对应。

PDE 问题的几何区域由三角形网格数据给出。关于三角形网格的表达方式可参阅 initmesh。

可以用多种方法给出 PDE 问题的系数 c。所有选项的完整表格可在 assempde 中找到。

解向量 u 的格式也在 assempde 中有所描述。

如果 c 依赖于时间 t，则选择参数 time (标量)被用于抛物型和双曲型 PDE 问题。

选择参数 sdl 用来限制在表格 sdl 中的子区域上进行计算。

pdegrad

计算 PDE 解的梯度。

格式：[ux,uy]=pdegrad(p,t,u)

[ux,uy]=pdegrad(p,t,u,sdl)

说明　[ux,uy]=pdegrad(p,t,u) 输出在每个三角形中心处 u 的梯度。

ux 共有 N 行；其中第 i 行是：$\frac{\partial u_i}{\partial x}$。

uy 共有 N 行；其中第 i 行是：$\frac{\partial u_i}{\partial y}$。

对于 t 中的每一个三角形，ux 和 uy 中都有一列与其对应。

PDE 问题的几何区域由三角形网格数据给出。关于三角形网格的表达方式可参阅 initmesh。

解向量 u 的格式也在 assempde 中有所描述。

选择参数 sdl 用来限制在表格 sdl 中的子区域上进行计算。

pdejmps

为自适应进行误差估计。

格式：errf=pdejmps(p,t,c,a,f,u,alfa,beta,m)

说明　errf=pdejmps(p,t,c,a,f,u,alfa,beta,m)计算为适应性而使用的指标函数的误差。

errf 的列数与三角形个数对应，行数与 PDE 方程的个数对应。p 和 t 是网格数据，详细情况参阅 initmesh。

c, a 和 f 是 PDE 的系数，详细情况参阅 assempde。c, a 和 f 必须展开，使得它们的列数与三角形的个数一致。

u 是解向量，详细情况参阅 assempde。

对于每个三角形 K 计算误差指标 $E(K)$的公式为

$$E(K)=\alpha\|h(f-au)\|_K+\beta\left\{\frac{1}{2}\sum_{\tau\in\partial K}h_\tau^2[n_\tau\cdot(c\nabla u_h)]^2\right\}^{\frac{1}{2}}.$$

此处 n_τ是边界 τ 的单位法向量，大括号内是穿过单元边界的流量跃度。范数是 K 单元上的 L_2 范数。

对于 t 中的每一个三角形，误差指标以列向量的形式存储在 errf 中。

pdesmech

计算结构力学张量函数。

格式：ux=pdesmech(p,t,c,u,'PropertyName',PropertyValue,...)

说明　ux=pdesmech(p,t,c,u,p1,v1,...)返回在每个三角形中心处的张量表达式。它是带有平面应力或应变条件的结构应用力学中的应力或应变。当应用了

PDE Tool GUI 的应用结构力学模式输出解，网格数据和 PDE 系数输出到主工作窗口后，利用 pdesmech 对解进行后处理工作。在平面应变模式中，为计算剪切力和应变，以及应变的 von Mises 的等效应力，必须明确地提供 Poisson 比。

正确的属性名和属性值对包括：

表 4-14

属性名	属性值{默认值}	说　明
tensor	ux|uy|vx|vy|exx|eyy|exy|sxx|syy|sxy|e1|e2|s1|s2|{von Mises}	张量表达式
application	{ps}|pn	平面应力|平面应变
nu	标量或字符串表达式{0.3}	Poisson 比

等效的应力表达式为

$$\mathrm{ux}=\frac{\partial u}{\partial x},\qquad \mathrm{uy}=\frac{\partial u}{\partial y},\qquad \mathrm{vx}=\frac{\partial v}{\partial x},\qquad \mathrm{vy}=\frac{\partial v}{\partial y},$$

exx——x 方向应变(ε_x)，

eyy——y 方向应变(ε_y)，

exy——剪切应变(γ_{xy})，

sxx——x 方向应力(σ_x)，

syy——y 方向应力(σ_y)，

sxy——剪切应力(τ_{xy})，

e1——第一主应变(ε_1)，

e2——第二主应变(ε_2)，

s1——第一主应力(σ_1)，

s2——第二主应力(σ_2)，

von Mises——von Mises 等效应力(平面应力条件用 $\sqrt{\sigma_1^2+\sigma_2^2-\sigma_1\sigma_2}$ ，平面应变条件用 $\sqrt{(\sigma_1^2+\sigma_2^2)(\nu^2-\nu+1)-\sigma_1\sigma_2(2\nu^2-2\nu-1)}$)。

【例 4.6-1】 在第三章中已求解过和讨论过“平面应力”问题。若此问题的解 u，网格数据 p 和 t，以及 PDE 的系数 c 已经输出到 MATLAB 主窗口中，则要计算 x 方向的应力，可键入：

```
sx=pdesmech(p,t,c,u,'tensor','sxx');
```

对于 Poisson 比等于 0.3 的平面应变问题，为计算应力的 von Mises 等效应

力，可键入如下命令：

```
mises=pdesmech(p,t,c,u,'tensor','von Mises',...
'application','pn','nu',0.3);
```

sptarn

求解一般稀疏矩阵特征值问题。

格式：[xv,lmb,iresult] = sptarn(A,B,lb,ub)

[xv,lmb,iresult] = sptarn(A,B,lb,ub,spd)

[xv,lmb,iresult] = sptarn(A,B,lb,ub,spd,tolconv)

[xv,lmb,iresult] = sptarn(A,B,lb,ub,spd,tolconv,jmax)

[xv,lmb,iresult] = sptarn(A,B,lb,ub,spd,tolconv,jmax,maxmul)

说明　[xv,lmb,iresult] = sptarn(A,B,lb,ub,spd,tolconv,jmax,maxmul)在区间[lb,ub]中求线束矩阵$(A-\lambda B)x=0$的特征值(称线性多项式$A_{ij}-\lambda B_{ij}$, $A-\lambda B$的矩阵为一个线束矩阵)。

A 和 B 是稀疏矩阵。lb 和 ub 分别是待求特征值的下界和上界。如果所有特征值都在 ub 的左边，则可设 lb=-inf；如果所有特征值都在 lb 的右边，则可设 ub=inf。lb 和 ub 中至少有一个必须是有限的。区间越小计算速度越快。复数情况下，把 lmb 的实部与 lb, ub 作比较。

xv 是特征向量，按照使范数 a*xv-b*xv*diag(lmb)较小的原则排序。lmb 是一个特征值。如果 iresult>=0，则算法是成功的，即区间内的所有特征值都被求出；如果 iresult<0，则算法不成功，即可能还有更多(未求出)的特征值，此时尝试缩短区间。

如果已知线束矩阵是正定对称矩阵，则 spd=1 (默认值是零)。

tolconv 是预期的相对精度，默认值是 100*eps，此处 eps 是机器精度。

jmax 是基底向量的最大数，由于算法需要 jmax*n 工作空间，所以，对于较小的计算机，jmax 取得小一点较为合适，否则设 jmax 为默认值(jmax=100)。当求出足够多的特征值时，算法较早地终止计算是正常的。

maxmul 是尝试 Arnoldi 算法运行的次数。至少与任一特征值的最大倍数一样大。如果 jmax 较小，则必须运行较多次 Arnoldi 算法。当需求出单位矩阵的所有特征值时，取默认值 maxmul=n。

算法

采用谱变换的 Arnoldi 算法。在区间端点 ub, lb 处选择变换，或者当区间端点都是有限的情况下，在区间(lb,ub)中随机地选一点进行变换。Arnoldi 算法运行的次数 j，取决于区间中特征值的个数，但当 j=min(jmax,n)时将停止运行。

然后，算法对这些已经被求出的特征值重新开始求更多正交的 Schur 向量。当在 lb < lmb ≤ ub 中不再求出更多的特征值时将终止运算。如果 jmax 较小，那么在求出几个特征值前，将会有好几次这种重新启动。当 jmax 至少比区间中特征值的个数大一个单位时，算法才能有效地运行，但需要重新启动许多次。首选较大的 jmax，运行 mul+1 次，mul 是区间中特征值的最大倍数。

注意：算法适用于非对称和对称线束(symmetric pencils)，但其精确度大致高于 Henrici 与正规值的偏离度的 tol 倍。当进行因式分解时，参数 spd 仅被用于在 symmmd 和 colmmd 之间进行选择。对于靠近谱的下界的对称矩阵，选择前者稍好。

可能发生的问题及对策：

(1) 如果收敛速度太慢，依以下步骤处理。

(2) 设置较小的区间 lb, ub。

(3) 设置较大的参数 jmax。

(4) 设置较大的参数 maxmul。

(5) 如果无法进行因式分解，可再次尝试将 lb 或 ub 设为有限的。然后选择在随机的点处而不寄希望于在特征值处进行变换。如果仍不能成功，则检查线束矩阵是否为奇异的。

(6) 如果运行不断进行，那么可能设置中有太多的特征值，尝试一下一个较小的值 maxmul=2，并检查是否能得到一些特征值。如果得到的特征值是负的，意味着并没有得到全部特征值。

(7) 如果内存不够，将 jmax 设为较小的值。

(8) 该算法是为特征值靠近实数轴而设计的。如果有一些靠近虚轴的特征值则可令 A=i*A 来试试。

(9) 当 spd=1 时，在 lb 处开始变换，对于正定对称矩阵这样做的优点是因式分解较快。如果 lb 比最小的特征值还大，这样做也没有关系，不过运行速度稍慢。

4.7 其他函数简介

pdegeom

创建偏微分方程定义区域的 M 文件。

格式：ne=pdegeom

d=pdegeom(bs)

[x,y]=pdegeom(bs,s)

说明 有两种方法创建偏微分方程定义区域的几何图形。一种是利用用户

图形界面(GUI)来创建描述几何图形的矩阵;另一种就是用工具箱中的命令来创建描述几何图形的 M 文件。当几何图形不是由直线、圆弧、椭圆弧及其组合而成的图形时，就只能用命令来创建 M 文件。

在创建描述几何图形的 M 文件时，必须遵循下面的法则:

(1) 所写的 M 文件必须能用上面的三种格式进行调用。

(2) 输入变量 bs 是指定的边缘线段，s 是相应线段弧长的近似(估计)值。

(3) 输出变量 ne=pdegeom 表示几何区域边界的线段数。

(4) 输出变量 d=pdegeom(bs)是一个区域边界数据的矩阵。

(5) d 的第 1 行是每条线段起始点的值。第 2 行是每条线段结束点的值。第 3 行是沿线段方向左边区域的标识值，如果标识值是 1，表示选定左边区域；如果标识值是 0，表示不选左边区域。第 4 行是沿线段方向右边区域的标识值，规则同上。

输出变量[x,y]=pdegeom(bs,s)是每条线段的起点和终点所对应的坐标。

【例 4.7-1】 下面是一个心形线 $r=2(1+\cos\phi)$ 所围图形的 M 文件。

把这条线分为 4 段：第 1 段的起点是 $\phi=0$，终点是 $\phi=\pi/2$；第 2 段的起点是 $\phi=\pi/2$，终点是 $\phi=\pi$；第 3 段的起点是 $\phi=\pi$，终点是 $\phi=3/2\pi$；第 4 段的起点是 $\phi=3/2\pi$，终点是 $\phi=2\pi$。

心形区域上图形的 M 文件 cardg.m 的内容是:

```
function [x,y]=cardg(bs,s)
    %CARDG是心形几何区域文件名
nbs=4;
if nargin==0
  x=nbs;
  return
end
dl=[0     pi/2  pi      3*pi/2
    pi/2  pi    3*pi/2  2*pi;
    1     1     1       1
    0     0     0       0];
if nargin==1
  x=dl(:,bs);
  return
end
x=zeros(size(s));
y=zeros(size(s));
[m,n]=size(bs);
```

```
if m==1 & n==1,
  bs=bs*ones(size(s)); % 扩展
elseif m~=size(s,1)|n~=size(s,2),
  error('bs must be scalar or of same size as s');
end
nth=400;
th=linspace(0,2*pi,nth);
r=2*(1+cos(th));
xt=r.*cos(th);
yt=r.*sin(th);
th=pdearcl(th,[xt;yt],s,0,2*pi);
r=2*(1+cos(th));
x(:)=r.*cos(th);
y(:)=r.*sin(th);
```

可以用 PDE Toolbox 中的其他命令来调用此文件。

在 MATLAB 的命令窗口键入下面的命令：

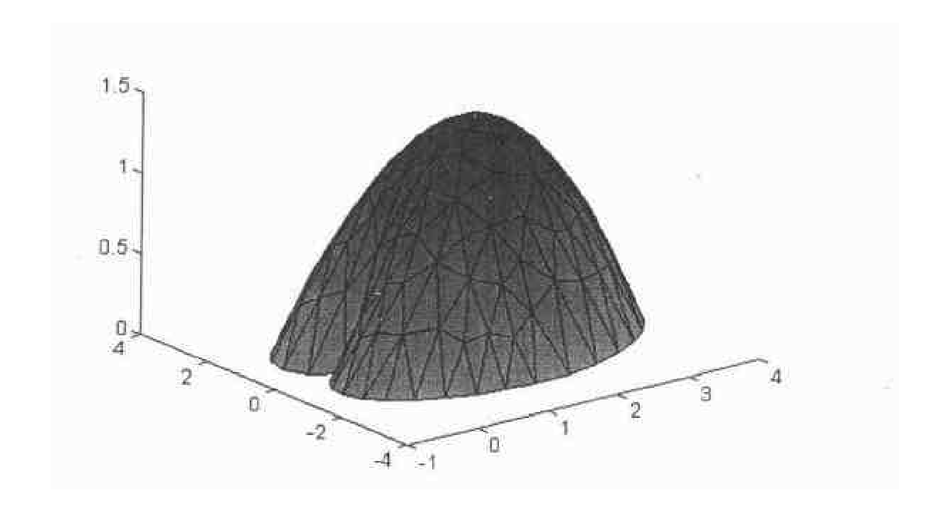
```
ne=cardg;
d=cardg([1,2,3,4]);
[x,y]=cardg([1,2,3,4],[2,1,1,2]);

ne
ne =
  4

d
d =
     0         1.5708    3.1416    4.7124
     1.5708    3.1416    4.7124    6.2832
     1.0000    1.0000    1.0000    1.0000
     0         0         0         0

x
x =
     0.4506    2.8663    2.8663    0.4506

y
y=
     2.3358    2.1694    2.1694    2.3358
```


1.5
1
0.5
0
4
2
0
-2
-4
-1
0
1
2
3
4

pdebound

创建边界条件的 M 文件。

格式：假设已创建了一个名为 pdebound 的边界条件 M 文件。那么调用此文件的格式为：[q,g,h,r]=pdebound(p,e,u,time)。

说明 边界条件的一般形式是：

$$\boldsymbol{hu} = \boldsymbol{r},$$

$$\boldsymbol{n} \cdot (\boldsymbol{c} \otimes \nabla \boldsymbol{u}) + \boldsymbol{qu} = \boldsymbol{g} + \boldsymbol{h}'\mu.$$

符号 $\boldsymbol{n} \cdot (\boldsymbol{c} \otimes \nabla \boldsymbol{u})$ 表示 $N \times 1$ 阶矩阵，其第 i 行为

$$\sum_{j=1}^{N} (\cos\alpha\, c_{i,j,1,1} \frac{\partial}{\partial x} + \cos\alpha\, c_{i,j,1,2} \frac{\partial}{\partial y} + \sin\alpha\, c_{i,j,2,1} \frac{\partial}{\partial x} + \sin\alpha\, c_{i,j,2,2} \frac{\partial}{\partial y}) u_j\ ,$$

其中 $\boldsymbol{n} = (\cos\alpha, \sin\alpha)$ 是外法线方向。有 M 个 Dirichlet 条件，且矩阵 $\boldsymbol{h}$ 是 $M \times N$ 阶的，其中 $M \geqslant 0$。广义的 Neumann 条件包含一个要计算的拉格朗日乘积因子 μ 的 $\boldsymbol{h}'\mu$ 以满足 Dirichlet 条件(编写此 M 文件的目的是要得到 q,g,h,和 r)。

如果 $M = 0$，可得广义的 Neumann 条件；如果 $M = N$，可得 Dirichlet 条件；如果 $M < N$ 可得混合边界条件。

边界条件的 M 文件[q,g,h,r]=pdebound(p,e,u,time)是在边缘 e 上算出 q,g,h,和 r 的值。

矩阵 p 和 e 是网格数据，且仅需求 e 是网格中边缘的子集。关于网格数据的表达方式可参阅 initmesh 中的解释。

输入变量 u 和 time 分别用于非线性求解器和时间步长的算法。如果相应的参数不能通过函数 assemb，那么 u 和 time 将是空矩阵。如果 time 是 NaN 且 q,g,h 和 r 中有任何一个是 time 的函数，那么 M 文件 pdebound 一定返回(一个)正确阶数的矩阵，但所有输出变量矩阵中的每个元素都将是 NaN。

解 u 由解向量 u 表示。详细的表达方式可参阅 assempde 中的描述。

q 和 g 必须包含每个边界中点的值，即 size(q)=[N^2 ne]，此处 N 是方程组的维数，ne 是 e 中的边界数。size(h)=[N ne]。对于 Dirichlet 条件，相应的值一定是零。

h 和 r 必须包含在每条边上的第 1 点的值，接着是在每条边上第 2 点的值。即 size(h)=[N^2 2*ne]，此处 N 是方程组的维数；ne 是 e 中边界数。size(r)=[N 2*ne]。当 $M < N$ 时，h 和 r 一定有 $N - M$ 行元素是零。

q 和 h 矩阵的元素以 q 和 h 的 MATLAB 矩阵的次序按列的方式存储。

例如：对于边界条件

$$(1, -1)\boldsymbol{u} = 2,$$

$$n(c\otimes\nabla u)+\begin{pmatrix}1 & 2\\ 2 & 0\end{pmatrix}u=\begin{pmatrix}3\\ 4\end{pmatrix}+h'\mu,$$

q,g,h 和 r 的存储方式为：

```
          1
q=[   ... 2 ...   ]
          2
          0

g=[   ... 3 ...   ]
          4

          1   1
h=[   ... 0 ... 0 ...   ]
         -1  -1
          0   0

r=[   ... 2 ... 2 ...   ]
          0   0
```

pdegrad

计算 PDE 解的梯度。

格式：[ux,uy]=pdegrad(p,t,u)

[ux,uy]=pdegrad(p,t,u,sdl)

说明　执行[ux,uy]=pdegrad(p,t,u)后返回在每个三角形中心的解 u 的梯度。ux 的第 i 行是 $\frac{\partial u_i}{\partial x}$；uy 的第 i 行是 $\frac{\partial u_i}{\partial y}$。

在 ux 和 uy 中，对于 t 中的每一个三角形都有一列。PDE 的几何区域由网格数据 p 和 t 给出。关于网格数据的详细表达方式可在 initmesh 中查到。

在 assempde 中有关于解向量 u 的格式。选择参数 sdl 控制计算表格 sdl 中的子区域。

第五章 有限元法和有限差分法

PDE Toolbox 求解偏微分方程采用的主要方法是有限元法。本章将对这一方法应用于几类典型偏微分方程问题作详细介绍。

5.1 椭圆型方程

PDE Toolbox 求解的基本椭圆型方程为

$$-\nabla\cdot(c\nabla u)+au=f,\quad \text{in}\ \ \Omega, \tag{5-1}$$

其中 Ω 是平面有界区域。c,a,f 以及未知函数 u 可以是定义在 Ω 上的复函数。边界条件可以是以下的情形之一：

- Dirichlet 条件：

$$hu=r,\quad \partial\Omega. \tag{5-2}$$

- 一般(广义)Neumann 条件：

$$\boldsymbol{n}\cdot(c\nabla u)+qu=g\ ,\quad \partial\Omega, \tag{5-3}$$

当 $q=0$ 时称为 Neumann 条件。

- 混合条件：在 $\partial\Omega$ 上部分是 Dirichlet 条件，部分是 Neumann 条件。

其中 $\boldsymbol{n}$ 是边界 $\partial\Omega$ 的单位法向量。g,q,h 和 r 是定义在 $\partial\Omega$ 上的函数。在有限元法中，Dirichlet 条件也称为本质边界条件，Neumann 条件称为自然边界条件。

有限元解实际上是微分方程弱形式的解（弱解或广义解）在有限维空间的投影。下面作详细讨论，为了简单起见，省去有关函数空间的严格定义。

1. 导出定解问题对应的弱形式

我们讨论方程(5-1)在一般 Neumann 条件(5-3)下的解。取任意试验函数 $v\in V$，乘(5-1)式的两边，并在 Ω 上积分：

$$\int_\Omega\left[-(\nabla\cdot(c\nabla u))v+auv\right]\mathrm{d}x=\int_\Omega fv\,\mathrm{d}x.$$

利用 Green 公式及条件(5-3)，可得

$$\int_\Omega\left[(c\nabla u)\cdot\nabla v+auv\right]\mathrm{d}x-\int_{\partial\Omega}(-qu+g)v\,\mathrm{d}s=\int_\Omega fv\,\mathrm{d}x.$$

于是，问题(5-1), (5-3)的弱形式（虚功方程）为：

求 $u\in V$，满足

$$\int_{\Omega}[(c\nabla u)\cdot\nabla v+auv-fv]\,\mathrm{d}x-\int_{\partial\Omega}(-qu+g)v\,\mathrm{d}s=0,\quad \forall v\in V. \tag{5-4}$$

(5-4)的解也称为问题(5-1), (5-3)的广义解(或弱解)。在一定意义下，问题(5-1), (5-3)与它的弱形式(5-4)是等价的。

如果问题是自共轭的，并且满足所谓椭圆型条件，那么问题(5-1), (5-3)也可以按最小势能原理导出泛函极小问题，这时它与虚功方程是等价的。

2. 区域剖分

由于三角形剖分在几何上有很大的灵活性，对边界 $\partial\Omega$ 的逼近较好，因此在 PDE Toolbox 中区域 Ω 一般作三角形网格剖分。一般而言，在作三角形剖分和节点编号时要注意以下几点：

- 单元顶点不能是相邻单元边上的内点。
- 尽量避免出现大的钝角、大的边长。
- 在 $u(x, y)$的梯度变化比较剧烈的地方，网格要加密。
- 单元编号可以任意，但节点编号的好坏，直接影响总刚度矩阵的带宽，应尽量使所有两个相邻节点编号之差的绝对值中的最大者愈小愈好。

3. 插值多项式的选取

在 PDE Toolbox 中采用的是单元上的线性函数：

$$u(x,y)=ax+by+c. \tag{5-5}$$

设节点 p_i 上 u 的值为 u_i，即 $u(x_i,y_i)=u_i$，$i=1,2,\cdots,N_p$，N_p 为节点数。任取单元 e，三个顶点为 p_i,p_j,p_m，记 $e=\overline{p_ip_jp_m}$。它们的顺序是逆时针的。为了使插值函数(5-5)在这三个顶点上分别取值 u_i,u_j,u_m，那么 a,b,c 应满足

$$ax_i+by_i+c=u_i,$$
$$ax_j+by_j+c=u_j,$$
$$ax_m+by_m+c=u_m.$$

解得 a,b,c，再代入(5-5)式，可得插值函数为

$$u(x,y)=N_i(x,y)u_i+N_j(x,y)u_j+N_m(x,y)u_m,$$

其中

$$N_i=\frac{1}{2\Delta_e}(a_ix+b_iy+c_i),$$

$$a_i=\begin{vmatrix} y_i & 1\\ y_m & 1\end{vmatrix},\quad b_i=-\begin{vmatrix} x_i & 1\\ x_m & 1\end{vmatrix},\quad c_i=\begin{vmatrix} x_i & y_i\\ x_m & y_m\end{vmatrix},$$

Δ_e为$\triangle p_i p_j p_m$的面积。N_j, N_m及$a_j, b_j, c_j, a_m, b_m, c_m$可由$N_i$及$a_i, b_i, c_i$的表达式按$i \to j \to m \to i$轮换得到。

设$\{\delta\}_e = [u_i, u_j, u_m]^T$，$[N] = [N_i, N_j, N_m]^T$，则在 e 上有

$$u = [u]\{\delta\}_e.$$

u 的梯度向量$\nabla u = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right)^T$可以表示为

$$\nabla u = \frac{1}{2\Delta_e}\begin{pmatrix} a_i & a_j & a_m \\ b_i & b_j & b_m \end{pmatrix}\{\delta\}_e = [B]\{\delta\}_e,$$

其中$[B]$是 2×3 常数矩阵。

4. 单元刚度矩阵、单元荷载向量的形成

设$\{\phi_i\}(i=1,2,\cdots,N_p)$为 V 的 N_p维子空间的基函数，按所剖分的单元将(5-4)式改成

$$\sum_{n=1}^{\mathrm{NE}} \iint_{e_n} [(c\nabla u)\cdot\nabla\varphi + au\varphi - f\varphi]\,\mathrm{d}x\,\mathrm{d}y = \sum_{n=1}^{\mathrm{NE}} \int_{r_n} (g\varphi - qu\varphi)\,\mathrm{d}s. \qquad (5\text{-}6)$$

这里 NE 是单元数，$r_n = \partial e_n \cap \partial\Omega$。

(5-6)式中的积分都只是在某一单元上求积，下面讨论每个单元上的积分计算。

$$\begin{aligned}
&\iint_e [(c\nabla u)\cdot\nabla\varphi + au\varphi]\,\mathrm{d}x\,\mathrm{d}y \\
&\quad = \iint_e \{c([B]\{\delta^*\}_e)^T([B]\{\delta\}_e) + a([u]\{\delta^*\}_e)^T([N]\{\delta\}_e)\}\,\mathrm{d}x\,\mathrm{d}y \\
&\quad = \{\delta^*\}_e^T[K]_e\{\delta\}_e,
\end{aligned}$$

$$\iint_e f\varphi\,\mathrm{d}x\,\mathrm{d}y = \iint_e ([N]\{\delta^*\}_e)^T f\,\mathrm{d}x\,\mathrm{d}y = \{\delta^*\}_e^T\{F\}_e = F_i,$$

$$\int_r qu\varphi\,\mathrm{d}s = \int_r q(\varphi)^T(u)\,\mathrm{d}s = \int_r q([N]\{\delta^*\}_e)^T([N]\{\delta\}_e)\,\mathrm{d}s = \{\delta^*\}_e^T[\overline{K}]_e\{\delta\}_e,$$

$$\int_r g\varphi\,\mathrm{d}s = \int_r (\varphi)^T(g)\,\mathrm{d}s = \int_r ([N]\{\delta^*\}_e)^T g\,\mathrm{d}s = \{\delta^*\}_e^T\{\overline{F}\}_e.$$

如果单元 e 是一边界单元，则单元刚度矩阵应为$[K]_e + [\overline{K}]_e$，单元荷载向量应为$\{F\}_e + \{\overline{F}\}_e$。这里三角单元的梯度和面积是通过行命令 pdetrg 来实现的。

5. 总刚度矩阵和总荷载向量的组装

将单元刚度矩阵和单元荷载向量表达式代入(5-6)式，一般来说，为了便于

叠加，先对$\{\delta\}_e,\{F\}_e,\{K\}_e$进行扩充，扩充为$N_p$维向量和$N_p\times N_p$阶方阵，然后“对号人座”进行叠加。在 PDE Toolbox 中这一组装过程通过命令行 eassempde 来实现。

6. 约束处理，求解方程组

对于 Neumann 条件，由于是自然边界条件，$\partial\Omega$上不需要满足任何约束条件，因此可立即得到线性代数方程组：

$$KU=F .$$

一旦形成K和F，在 MATLAB 环境下立即可解出节点近似解的向量u。

如果是 Dirichlet 边界条件，还需要对边界节点进行约束处理，然后再解方程组。

5.2　抛物型方程

下面说明抛物型方程如何简化成椭圆型方程来求解。这是通过 PDE Toolbox 函数 parabolic 完成的。

考虑

$$d\frac{\partial u}{\partial t}-\nabla\cdot(c\nabla u)+au=f ,\quad \text{in}\,\Omega, \tag{5-7}$$

初值为

$$u(x,0)=u_0(x),\quad x\in\Omega . \tag{5-8}$$

边界条件类似椭圆边值问题的提法，这里仅讨论 Neumann 条件：

$$\boldsymbol{n}\cdot(c\nabla u)+qu=g,\quad \partial\Omega . \tag{5-9}$$

对于热传导方程(5-7)可以写成

$$\rho C\frac{\partial u}{\partial t}-\nabla\cdot(k\nabla u)+h(u-u_\infty)=f , \tag{5-10}$$

表示热量向环境扩散，其中ρ是密度，C是比热，k是导热系数，h是薄层传热系数，u_∞是环境温度，f是热源。

如果系数与时间无关，方程就是标准的椭圆型方程：

$$-\nabla\cdot(c\nabla u)+au=f .$$

对Ω作三角形网格剖分，对于任意给定$t\geqslant 0$，PDE 的解按有限元法的基底可以展开成

$$u(x,t)=\sum_i u_i(t)\varphi_i(x) . \tag{5-11}$$

将展开式代入方程(5-7)，两边乘以试验函数 φ_i，然后在 Ω 上积分，利用 Green 公式和边界条件(5-9)，可得

$$\sum_i \int_\Omega d\varphi_j\varphi_i \mathrm{d}x \frac{\mathrm{d}u_i(t)}{\mathrm{d}t} + \sum_i \Big(\int_\Omega \nabla\varphi_j\cdot(c\nabla\varphi_i) + a\varphi_j\varphi_i \mathrm{d}x + \int_{\partial\Omega} q\varphi_j\varphi_i \mathrm{d}s\Big) v_i(t)\mathrm{d}x$$
$$= \int_\Omega f\varphi_j \mathrm{d}x + \int_{\partial\Omega} g\varphi_j \mathrm{d}s\ ,\ \forall j. \tag{5-12}$$

上式可以写成大型的线性稀疏的常微分方程组:

$$M\frac{\mathrm{d}U}{\mathrm{d}t} + KU = F. \tag{5-13}$$

这就是所谓的线性半离散化方法。

再解 ODE (5-13)的初值问题，初值为

$$U_i(0) = u_0(x_i).$$

得每一个节点 x_i、任一时刻 t 的 ODE 解。这里 K 和 F 是原边界条件下椭圆型方程

$$-\nabla\cdot(c\nabla u) + au = f\ ,\quad \text{in}\ \Omega$$

的刚度矩阵和荷载向量。M 是质量矩阵。

当边界条件是与时间有关的 Dirichlet 条件 $hu = r$ 时，F 项则包括 h 和 r 的时间导数，它可以用有限差分法求解。常微分方程组是病态的，这时需作显式时间积分。

由于稳定性要求时间间隔很小，而隐式解由于每一时间段都要求解椭圆型方程，从而求解非常缓慢。常微分方程组的数值积分，可以由 MATLAB 中 suite 函数完成。对于这类问题它是有效的。

5.3 双曲型方程

类似于抛物型方程的有限元法，可以解双曲型问题：

$$d\frac{\partial^2 u}{\partial t^2} - \nabla\cdot(c\nabla u) + au = f\ ,\quad \text{in}\ \Omega\ .$$

初值为 $u(x,0) = u_0(x)$，$\dfrac{\partial u}{\partial t}(x,0) = v_0(x)$，$x\in\Omega$，边界条件同上。

特别地，对于波动方程 $\dfrac{\partial^2 u}{\partial t^2} - c\Delta u = 0$，波速为 $\sqrt{c}$。

对区域 Ω 作三角形网格剖分，与抛物型方程处理方法一样，可以得到二阶

常微分方程组:

$$M\frac{\mathrm{d}^2V}{\mathrm{d}t^2}+KV=F\ ,$$

初值为$V_i(0)=u_0(x_i)$, $\frac{\mathrm{d}}{\mathrm{d}t}V_i(0)=V_0(x_i)$,　$\forall i$.

K和M是刚度矩阵和质量矩阵。

PDE Toolbox 中提供的求解双曲型方程的命令函数是 hyperbolic。

5.4　特征值方程

在 PDE Toolbox 中求解的基本特征值问题是

$$-\nabla\cdot(c\nabla u)+au=\lambda du\ ,$$

其中λ是未知复数。在固体力学中方程描述薄膜振动的问题,在量子力学中λ描述势阱 $a(x)$中有界定态的能级。数值解包括方程的离散和代数特征值问题的求解。首先考虑离散化。按有限元基底将u展开，两边同乘基函数，再在Ω上作积分，可以得到广义特征值方程：

$$KU=\lambda MU\ ,$$

其中对应于右端项的质量矩阵的元素为

$$M_{i,j}=\int_\Omega d(x)\varphi_j(x)\varphi_i(x)\mathrm{d}x.$$

利用命令函数 assema（单元积分贡献的装配）生成。在通常情况下，当函数$d(x)$为正时,质量矩阵M为正定对称矩阵。同样,当$c(x)$是正的而且在 Dirichlet 边界条件下，刚度矩阵K也是正定的。

广义特征值问题：

$$KU=\lambda MU.$$

利用 Arnoldi 算法进行移位和求逆矩阵，直到所有的特征值都落在用户事先确定的区间。在此，求解的详细过程略去。

PDE Toolbox 中提供的求解特征值问题的命令函数是 pdeeig。

5.5　非线性方程

由于实际计算的许多问题是非线性的，因此在 PDE Toolbox 的 assempde 函数上建立了一个非线性求解器。

采用的基本方法是 Gauss-Newton 迭代法。求解的非线性方程为

$$r(u)=-\nabla\cdot(c(u)\nabla u)+a(u)u-f(u)=0,\quad \text{in}\quad \Omega.$$

一般 Neumann 条件为

$$\boldsymbol{n}\cdot(c\nabla u)+qu=g,\quad \text{on}\quad \partial\Omega.$$

用有限元求解的是 $r(u)$=0 的弱解。先作 $u(x)=\sum v_j\varphi_j$，让方程的两边同乘以任意试验函数 φ_i，然后在 Ω 上作积分，再利用 Green 公式和边界条件即可得

$$0=\rho(v)=\sum_j\Big(\int_\Omega(c(x,v)\nabla\varphi_j(x))\cdot\nabla\varphi_i(x)+a(x,v)\varphi_j(x)\varphi_i(x))\mathrm{d}x+$$

$$\int_{\partial\Omega}q(x,v)\varphi_j\varphi_i(x)\mathrm{d}s)v_j-\int_\Omega f(x,v)\varphi_i(x)\mathrm{d}x-\int_{\partial\Omega}g(x,v)\varphi_i(x)\mathrm{d}s,$$

这样，残量向量 $\rho(v)$ 容易求出：

$$\rho(v)=(K+M+Q)v-(F+G),$$

其中矩阵 K, M, Q 及向量 F, G 是由单元组装而成的。

假设一个解的初值为 $v^{(n)}$,并且与精确解十分接近，那么下一步改进的近似解 $v^{(n+1)}$可以由如下线性化问题解得：

$$\frac{\partial\rho(v^{(n)})}{\partial v}(v^{(n+1)}-v^{(n)})=-\alpha\rho(v^{(n)}),$$

其中 α 是一个正数，只要 $\alpha>0$ 充分小，则有

$$\left\|\rho(v^{(n+1)})\right\|<\left\|\rho(v^{(n)})\right\|,$$

并且 $p_n=(\frac{\partial\rho(v^n)}{\partial v})^{-1}\rho(v^{(n)})$ 称为 $\|\rho(v)\|$ 的下降方向，这里$\|\ \|$取 L_2 范数，那么迭代公式可以写成

$$v^{(n+1)}=v^{(n)}+\alpha p_n,$$

其中 $\alpha\leqslant 1$ 应尽可能大以保证一个适当的下降速度。

由于 Gauss-Newton 迭代法是局部收敛的，也就是说对于初值 $v^{(0)}$在靠近解的附近才能保证收敛。一般而言，$v^{(0)}$可能不在收敛域内，为此，在构造 α 时采用了一种阻尼 Gauss-Newton 法，也称为 Armijo-Goldsteim line 搜索法。这种方法是在数列 $1,\frac{1}{2},\frac{1}{4},\cdots$ 中选出最大的阻尼系数 α，使其满足如下不等式：

$$\left\|\rho(v^{(n)})\right\|-\left\|\rho(v^{(n)}+\alpha p)_n\right\|\geqslant\frac{\alpha}{2}\left\|\rho(v^{(n)})\right\|.$$

这样能保证残余范数至少按 $1-\frac{\alpha}{2}$ 的速度下降。这种方法的重要一点在于，当 $v^{(n)}$ 趋于解时，$\alpha\to 1$，同时收敛速度也在增加。如果 $\rho(v)=0$ 有解，那么这

种算法最终能达到标准的 Newton 迭代的二次收敛速度。

PDE Toolbox 中提供的解非线性方程的命令函数是 pdenonlin。

5.6 用有限元法求解的应用实例

本小节将依据有限元法原理，用 MATLAB 语言而不用 PDE Toolbox GUI 编写一段应用程序。这样做可以进一步学习有限元法的原理及其求解过程。另外，可以看到 PDE Toolbox 的强大功能。

【例 5.6-1】 图 5-1 为一个同轴传输电缆的截面示意图。两个同芯长方形导体之间充满了线性介质。假设导体之间加有 10 V 的直流电压，内导体接地，导体之间有密度为 ρ 的电荷，传输线的长度充分长，则可认为电场在传输线各个截面上的分布相同，因此只需求解电场在某个截面上的分布，即求出电场在导体间的分布，从而检查出绝缘材料的工作情况。

由于该问题的几何形状、介质都关于 x 轴和 y 轴对称，所以只需求解整个截面的四分之一即可，如图 5-2。

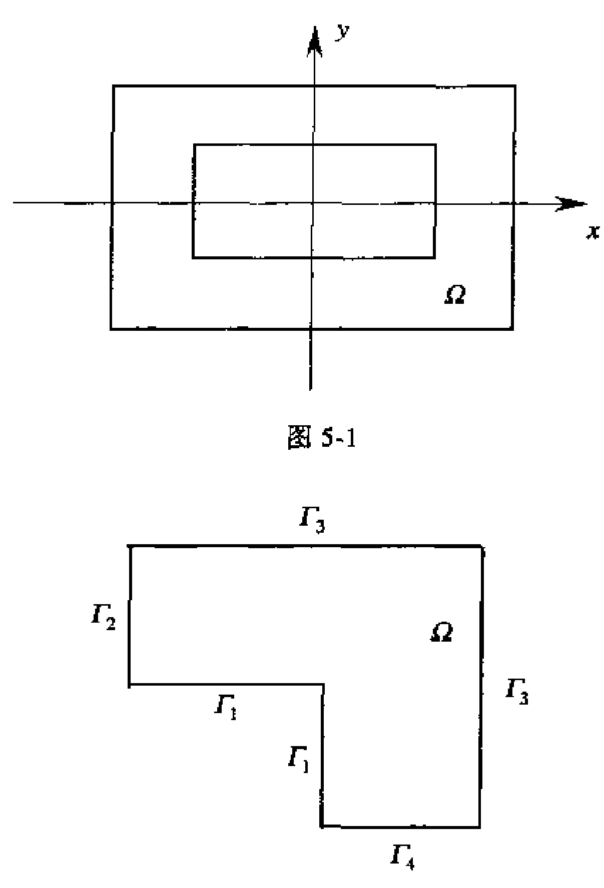


图 5-1

图 5-2

由静电场的理论知，此问题的数学模型可由下式描述：

$$\nabla^2\phi = -q, \quad (x,y)\in \Omega,\ q = \frac{\rho}{\varepsilon},$$

$$\phi\big|_{\Gamma_1} = 0, \quad \phi\big|_{\Gamma_3} = 10,$$

$$\frac{\partial\phi}{\partial n}\Big|_{\Gamma_2} = \frac{\partial\phi}{\partial n}\Big|_{\Gamma_4} = 0 .$$

附录二正是求解此问题的 MATLAB 程序。可以看到，尽管没有用 PDE Toolbox GUI 求解此题，但仍能显示出 MATLAB 语言较其他计算机语言简单、方便和绘图功能强大的特点。对于此程序，有以下几点说明：

- 用有限元法解题，需经前处理、组装刚度矩阵、解线性方程组、后处理这 4 个过程。本程序即是按这 4 个过程编写的。
- 为了便于形成刚度矩阵和求解线性方程组，本例将求解区域分为两个子区域 G_1, G_2，如图 5-3。

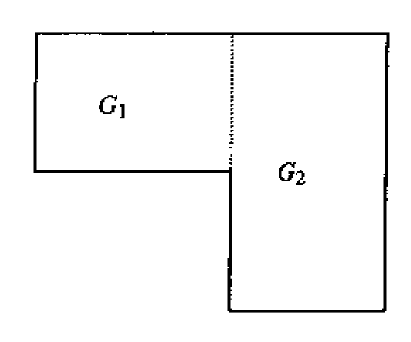


图 5-3

- 本程序需分别输入区域 G_1 和 G_2 的 x 和 y 的取值范围。还需输入划分区域的横向线和纵向线根数。本例中 G_1 的 x 的取值范围是[0,1]，y 的取值范围是[1,3]；G_2 的 x 的取值范围是[1,2]，y 的取值范围是[0,3]。划分 G_1 的横向线根数为 6，纵向线根数为 7，划分 G_2 的横向线根数为 6，纵向线根数为 10。显然，这样处理程序更具灵活性和通用性。可能对有的 MATLAB 版本不能运行此程序，如果是这样，将 input 语句改为赋值语句问题即可解决。
- 本程序的节点编号和单元编号均遵循从下到上、从左到右的原则。w1,w2 分别为 G_1 和 G_2 相应节点处的解。
- 本程序在绘制网格图时，主要应用了两个函数命令：sparse 和 gplot。下面介绍这两个命令的使用方法。

sparse 的调用格式为

```
SP=sparse(I, J, S, m, n, nzmax)
```

I, J, S 这三个向量的长度相同。S 是按列排列的所有非零元素构成的主对角线向量。I, J 分别是非零元素的行下标和列下标向量。

m, n 分别是生成稀疏矩阵 SP 的行、列维数。

nzmax 是用来为非零元素指定存储空间的正整数(它一定不小于非零元素的总数)。

调用格式中的 6 个输入变量在一定条件下可以缺省。

如果 S = sparse(i,j,s,m,n)，则将使 nzmax = length(s)。

如果 S = sparse(i,j,s)，则 m = max(i) , n = max(j)。

如果 S = sparse(m,n)，则简化成 SPARSE([],[],[],m,n,0)。

如果 S= sparse(A)，则可变为[i,j,s] = find(A);S = sparse(i,j,s)。

例如，将矩阵$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \\ 4 & 3 & 0 \\ 0 & 0 & 0 \end{pmatrix}$变成稀疏矩阵：

输入

```
A=[ 0 1 0;0 0 2; 0 0 0;4 3 0; 0 0 0];
SP=sparse(A)
```

结果是

```
SP =
    (4,1)        4
    (1,2)        1
    (4,2)        3
    (2,3)        2
```

输入：

```
gplot
```

MATLAB 可用稀疏矩阵所提供的信息通过 gplot 绘制网格图。假定有如图 5-4 那样的简单网格，那么如何用 MATLAB 稀疏矩阵表示上面的网格信息呢?

这里需要两种信息，第 1 种信息为顶点之间的连接关系，它用一个邻接矩阵 A 表示，假定：

A(i,i)=同节点 i 相连的节点数，也就是邻接节点数。例如，A(1,1)=3。

A(i,j)=0（节点 i 与 j 不相连）或-1（节点 i 与 j 相连）。

第 2 种信息为节点坐标信息向量 xy，规定：xy(i)=[x(i) y(i)]为第 i 个节点的坐标，那么上面的简单网格就可以表示为

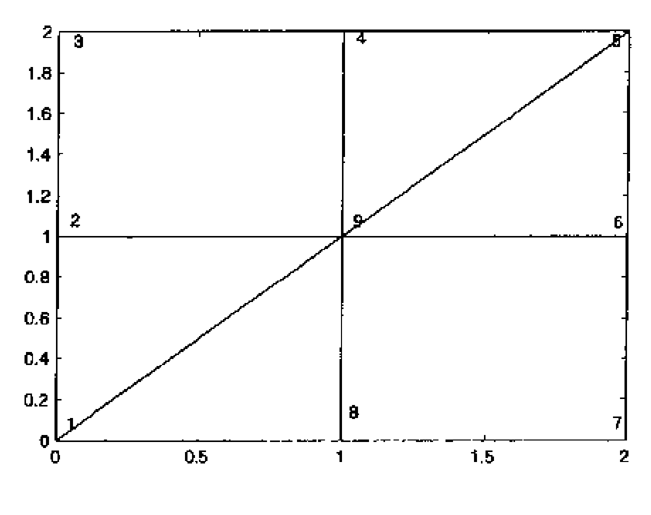


图 5-4

```
A= [1 1 3;1 2 -1;1 9 -1;1 8 -1;
    2 2 3;2 1 -1;2 3 -1;2 9 -1
    3 3 2;3 4 -1;3 2 -1;
    4 4 3;4 3 -1;4 5 -1;4 9 -1;
    5 5 3;5 4 -1;5 6 -1;5 9 -1;
    6 6 3;6 5 -1;6 7 -1;6 9 -1;
    7 7 2;7 6 -1;7 8 -1
    8 8 3;8 7 -1;8 1 -1;8 9 -1
    9 9 6;9 1 -1;9 2 -1;9 4 -1;9 5 -1;9 6 -1;9 8 -1]
```

节点坐标矩阵为

```
xy= [0 0;1 0;2 0;2 1;2 2;1 2;0 2;0 1;1 1]
```

上面是数据的输入格式，MATLAB 全部采用稀疏矩阵表示，因此，要把它们转化为 MATLAB 的稀疏矩阵格式，转化格式的方法如下：

```
A=spconvert(A)
A=
   (1,1)                 3
   (2,1)                -1
   (8,1)                -1
   (9,1)                -1
   (1,2)                -1
   (2,2)                 3
   (3,2)                -1
   (9,2)                -1
```

```
(2,3)        -1
(3,3)         2
(4,3)        -1
(3,4)        -1
(4,4)         3
(5,4)        -1
(9,4)        -1
(4,5)        -1
(5,5)         3
(6,5)        -1
(9,5)        -1
(5,6)        -1
(6,6)         3
(7,6)        -1
(9,6)        -1
(6,7)        -1
(7,7)         2
(8,7)        -1
(1,8)        -1
(7,8)        -1
(8,8)         3
(9,8)        -1
(1,9)        -1
(2,9)        -1
(4,9)        -1
(5,9)        -1
(6,9)        -1
(8,9)        -1
(9,9)         6
```

其余为 0 的元素，表示节点之间不连接。

经过上面的转化，可把网格图形数据转化为 MATLAB 的稀疏矩阵。反过来，用这样的矩阵可画出所要的网格图。这只需按规则写好稀疏矩阵，再调用 gplot(A,[x y],'k')即可。

将附录二中程序存为 M 文件，在 MATLAB 命令窗口中运行此文件即可得到如下结果，图形显示如图 5-5 所示。

```
w1=
   10.0000   10.0000   10.0000   10.0000   10.0000   10.0000
    9.2302    9.2420    9.2773    9.3358    9.4162    9.5158
```

```
    8.1727    8.1964    8.2681    8.3883    8.5570    8.7708
    6.7610    6.7956    6.9016    7.0842    7.3503    7.7029
    4.9345    4.9745    5.0997    5.3262    5.6807    5.1961
    2.6637    2.6945    2.7944    2.9904    3.3420    3.9749
    0         0         0         0         0         0
w2 =
   10.0000   10.0000   10.0000   10.0000   10.0000   10.0000
    9.5158    9.6293    9.7483    9.8614    9.9528   10.0000
    8.7708    9.0208    9.2910    9.5607    9.8059   10.0000
    7.7029    8.1336    8.6132    9.1061    9.5786   10.0000
    6.1961    6.8887    7.6758    8.4859    9.2711   10.0000
    3.9749    5.1590    6.4525    7.7125    8.8976   10.0000
    0         2.9032    5.1002    6.9252    8.5312   10.0000
    0         2.3869    4.5590    6.5243    8.3234   10.0000
    0         2.2448    4.3698    6.3632    8.2334   10.0000
    0         2.2133    4.3240    6.3217    8.2095   10.0000
```

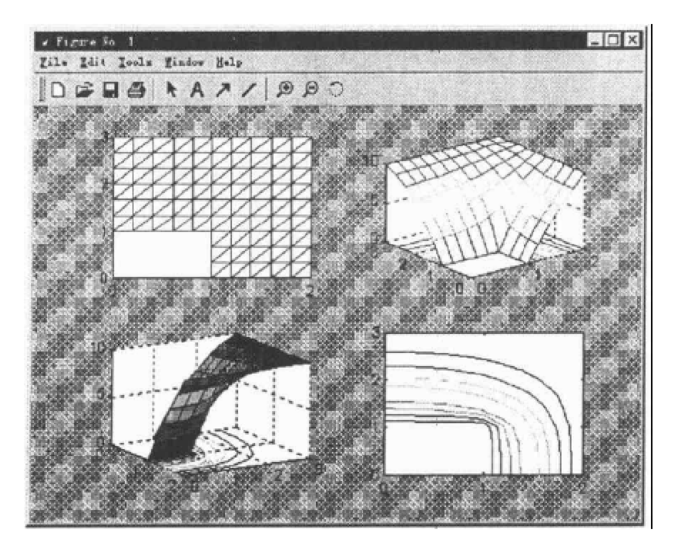


图 5-5

5.7 典型方程的有限差分法简介

5.7.1 矩形区域椭圆型方程的差分格式

为讨论简单起见，考虑定义于区域 $\Omega=\{0<x<a,\ 0<y<b\}$ 上的自共轭椭圆型方程:

$$-\left[\frac{\partial}{\partial x}(p\frac{\partial u}{\partial x})+\frac{\partial}{\partial y}(p\frac{\partial u}{\partial y})\right]+qu=f\,, \tag{5-14}$$

首先，对区域Ω作均匀网格剖分：$h_1=a/M$、$h_2=b/N$.区域内的节点为(x_i,y_j)，$i=1,2,\cdots,M-1$，$j=1,2,\cdots,N-1$.假设解充分光滑，则沿x和y方向分别用中心差商代替导数，从而在(x_i,y_j)点上有

$$\frac{\partial}{\partial x}(p\frac{\partial u}{\partial x})=\frac{1}{h_1}(p_{i+\frac{1}{2},j}\frac{u_{i+1,j}-u_{ij}}{h_1}-p_{i-\frac{1}{2},j}\frac{u_{ij}-u_{i-1,j}}{h_1})+O(h_1^2)\,,$$

$$\frac{\partial}{\partial y}(p\frac{\partial u}{\partial y})=\frac{1}{h_2}(p_{i,j+\frac{1}{2}}\frac{u_{i,j+1}-u_{ij}}{h_2}-p_{i,j-\frac{1}{2}}\frac{u_{ij}-u_{i,j-1}}{h_2})+O(h_2^2)\,,$$

其中u_{ij}表示$u(x_i,y_j)$的近似值，$p_{ij}=p(x_i,y_j)$。以下记号类推。于是略去局部截断误差$O(h_1^2+h_2^2)$后，可得逼近(5-14)的五点差分格式为

$$-(\alpha_1u_{i+1,j}+\alpha_2u_{i,j+1}+\alpha_3u_{i-1,j}+\alpha_4u_{i,j-1})+\alpha_0u_{ij}=f_{ij}\,. \tag{5-15}$$

上式中的系数是

$$\alpha_0=\sum_{k=1}^{4}\alpha_k+q_{ij}\,,\quad \alpha_1=\frac{1}{h_1^2}p_{i+\frac{1}{2},j}\,,$$

$$\alpha_2=\frac{1}{h_2^2}p_{i,j+\frac{1}{2}}\,,\quad \alpha_3=\frac{1}{h_1^2}p_{i-\frac{1}{2},j}\,,\quad \alpha_4=\frac{1}{h_2^2}p_{i,j-\frac{1}{2}}\,.$$

对于 Poisson 方程的 Dirichlet 问题：

$$-\Delta u=f(x,y)\,,\quad \text{in }\ \Omega\,, \tag{5-16}$$

$$u=\alpha(x,y)\,,\quad \text{on }\ \partial\Omega\,, \tag{5-17}$$

如果用正方形网格：$h_1=h_2=h$，并且Ω为正方形区域，则立即得到求解 Poisson 方程的差分格式为

$$-u_{i,j+1}-u_{i-1,j}+4u_{ij}-u_{i+1,j}-u_{i,j-1}=h^2f_{ij}\,. \tag{5-18}$$

如果按如下自然次序排列定义向量：

$$u_h=[u_{11},u_{21},\cdots,u_{N-1,1},\cdots,u_{1,N-1},u_{2,N-1},\cdots,u_{N-1,N-1}]^{\mathrm{T}}\,,$$

则利用边界条件可以得到如下$(N-1)^2$阶代数方程组：

$$\frac{1}{h^2}Hu_h=g\,,$$

其中，

$$H=\begin{pmatrix} B & -I & & \\ -I & B & -I & \\ & \ddots & \ddots & \\ & & -I & B \end{pmatrix},\quad B=\begin{pmatrix} 4 & -1 & & \\ -1 & 4 & -1 & \\ & \ddots & \ddots & \\ & & -1 & 4 \end{pmatrix}$$

为 $N-1$阶矩阵。向量 g 的元素依赖于边值 $\alpha(x,y)$ 及右端项 $f(x,y)$，它们都是已知的。这里 H 是对称正定矩阵。

5.7.2 热传导方程的差分格式

考虑简单的常系数热传导方程：

$$\frac{\partial u}{\partial t}-a\frac{\partial^2 u}{\partial x^2}=f,\quad a>0, x\in(0,1), t\in(0,T), \tag{5-19}$$

$$u(x,0)=\phi(x),\quad x\in[0,1], \tag{5-20}$$

$$u(0,t)=u(1,t)=0,\quad t\in[0,T]. \tag{5-21}$$

古典显格式

取网格的空间步长为 $h=\frac{1}{N}$，时间步长为 τ。在 (x_j,t_k) 处，时间用向前差商，空间用中心差商近似，立即得到古典显格式：

$$\frac{u_{j,k+1}-u_{jk}}{\tau}-\frac{a}{h^2}(u_{j+1,k}-2u_{jk}+u_{j-1,k})=f_{jk}, \tag{5-22}$$

$j=1,2,\cdots,N-1, k=0,1,\cdots,[T/\tau]-1$。进一步将显格式写成

$$u_{j,k+1}=u_{jk}+ra(u_{j+1,k}-2u_{jk}+u_{j-1,k})+\tau f_{jk}, \tag{5-23}$$

其中 $r=\tau/h^2$ 称为网格比。利用(5-23)及(5-20)，(5-21)在节点上就可以依次求出 $k=1,2,\cdots,[T/\tau]$ 各时间层上的 u_{jk}。

这个格式的局部截断误差为 $O(\tau+h^2)$。当且仅当 $ra\leqslant\frac{1}{2}$ 时格式是稳定的。

古典隐格式

在 (x_j,t_{k+1}) 处时间用向后差商，空间用中心差商近似，立即得到古典隐格式：

$$\frac{u_{j,k+1}-u_{jk}}{\tau}-\frac{a}{h^2}(u_{j+1,k+1}-2u_{j,k+1}+u_{j-1,k+1})=f_{j,k+1}, \tag{5-24}$$

$j=1,2,\cdots,N-1,\quad k=0,1,\cdots,[\frac{T}{\tau}]-1$。局部截断误差为 $O(\tau+h^2)$。当从已知 k 层求 $k+1$ 层时，需要求解一个 $N-1$ 阶方程组，但这个方程组是三对角、强对角占

优的，可以用追赶法求解。古典隐格式的稳定性对网格比没有要求，即为绝对稳定的。

二维的热传导方程定解问题

$$\frac{\partial u}{\partial t}=\Delta u,\quad x,y\in(0,1),\ t\in(0,T), \tag{5-25}$$

$$u(x,y,0)=\phi(x,y),\quad x,y\in[0,1], \tag{5-26}$$

$$u(0,y,t)=u(1,y,t)=u(x,0,t)=u(x,1,t)=0,\quad t\in[0,T],\ x,y\in[0,1]. \tag{5-27}$$

取两个空间方向步长均为 h, $h=1/N$，则古典显格式为

$$u_{ij}^{k+1}=(1-4r)u_{ij}^{k}+r(u_{i+1,j}^{k}+u_{i-1,j}^{k}+u_{i,j+1}^{k}+u_{i,j-1}^{k}). \tag{5-28}$$

其局部截断误差为 $O(\tau+h^2)$，$r=\tau/h^2$。当且仅当 $r\leqslant 1/4$ 时格式是稳定的。

方程(5-25)的古典隐格式是

$$\frac{u_{ij}^{k+1}-u_{ij}^{k}}{\tau}=\frac{1}{h^2}(u_{i+1,j}^{k+1}-2u_{ij}^{k+1}+u_{i-1,j}^{k+1}+u_{i,j+1}^{k+1}-2u_{ij}^{k+1}+u_{i,j-1}^{k+1}), \tag{5-29}$$

$i,j=1,2,\cdots,N-1$。其局部截断误差为 $O(\tau+h^2)$，网格比为 $r=\tau/h^2$。隐格式对任何网格比都是稳定的。

5.8　用差分方法求解的例子

【例 5.8-1】　横截面为矩形的无限长槽由 3 块接地导体板构成，槽的盖板电压为 $V_0=100$，其侧壁与底部电压为 0，求矩形槽中的电位分布。这一问题归结为如下定解问题：

$\Delta u=0$，$\Omega=\{0<x<17,0<y<10\}$，

$u(0,y)=u(17,y)=u(x,0)=0$，

$u(x,10)=100$.

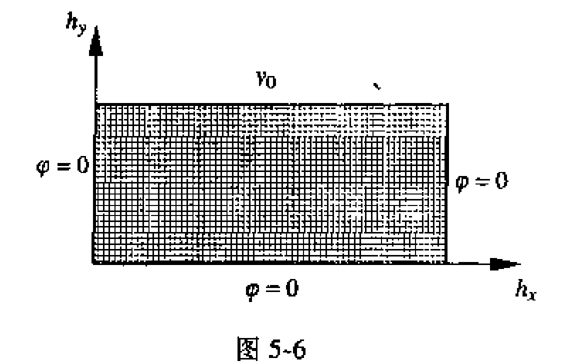


图 5-6

求解程序如下：

```
v0=100;hx=17;hy=10;
v1=zeros(hy,hx);
v1(hy,:)=ones(1,hx)*v0;
  %上边界值
v1(2:hy-1,2:hx-1)=ones(hy-2,hx-2);  %赋初值
v2=zeros(hy,hx);  maxt=1;t=0;  %初始化
v2(hy,:)=v1(hy,:);  %上边界值
while (maxt>0.1)  %由v1迭代,算出v2,迭代精度为0.1
```

```
for i=2:hy-1
    for j=2:hx-1
    v2(i,j)=(v1(i,j-1)+v1(i,j+1)+v1(i-1,j)+...
    v1(i+1,j))/4;  %用五点格式差分法
    t=v2(i,j)-v1(i,j);
    maxt=0;
    if(t>maxt) maxt=t;end
      end
    end
  v1=v2;
end
subplot(1,2,1),surf(v2)  %如图5-7所示,左图为曲面图
axis([0,17,0,10,0,100])
  subplot(1,2,2),
  contour(v2);  %画等值线
  hold on;  %保屏
  x=1:1:17;y=1:1:10;
  [xx,yy]=meshgrid(x,y);  %形成栅格
[Gx,Gy]=gradient(v2,0.6,0.6);  %计算梯度
quiver(xx,yy,Gx,Gy, 'r'),  %画矢量图
hold off
```

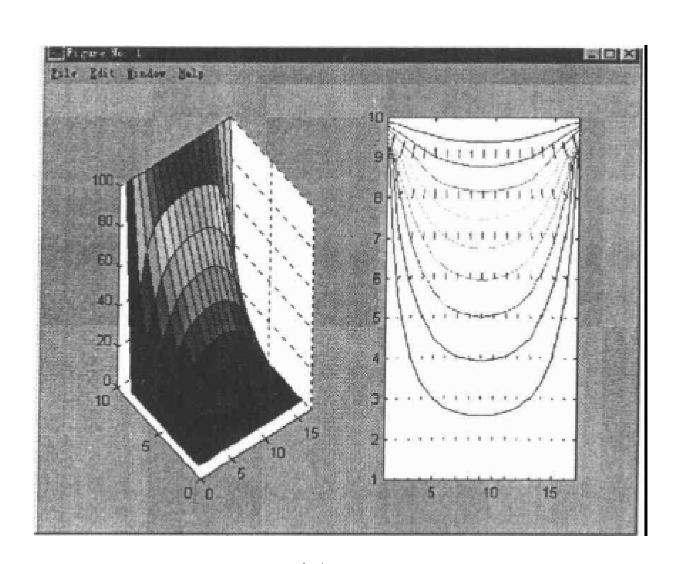


图 5-7

第六章　常微分方程及方程组的解法

除了偏微分方程工具箱以外，MATLAB 还提供了求常微分方程数值解和解析解的命令。本章将对此作简单介绍。

6.1　求常微分方程及方程组的初值问题的数值解

在工程计算中，求常微分方程或常微分方程组的数值解很常见，而且非常重要。MATLAB 提供了求常微分方程或常微分方程组的数值解的函数命令。表 6-1、6-2 列举了这些函数命令及其说明。

表 6-1　非刚性方程的函数

ode23	基于 Runge-Kutta(2,3)法，单步法，低阶，精度较低，速度较快。
ode45	基于 Runge-Kutta(4,5)法，单步法，中阶，精度较高，速度较慢。
ode113	采用 Adams-Bashforth-Moulton 法，各(可变)阶方法，多步法。

表 6-2　刚性方程的函数

ode15s	采用多步法，精度较低。
ode23s	采用二阶改进的 Rosenbrock 法，单步法，速度较快。

MATLAB 提供的这些函数命令适用于一阶的常微分方程的初值问题：

$$y' = f(x, y), \quad a \leqslant x \leqslant b,$$

$$y(x_0) = y_0.$$

对于高阶微分方程，必须先用替换法化为形如 $y' = f(x,y)$的一阶微分方程组，即“状态方程”，再加以使用。MATLAB 提供的这些函数命令，均放在\matlabR12\

toolbox\matlab\funfun 目录下。非刚性方程的函数命令的调用格式为

[X,Y]=odeN('odex',[t0,tf,]y0,tol,trace)

odeN 可以是 ode23, ode45, ode113, ode15s, ode23s 中的任意一个命令。

第 1 个输入参变量'odex'是定义 $f(x, y)$的函数文件名。该函数文件必须以 $y'=f(x,y)$为输出，以 x, y 为输入参变量，次序不能颠倒。

输入参变量 t0 和 tf，分别是积分的初值和终值。

输入参变量 y0 是初始状态列向量。

第 5 个输入参变量 tol 控制解的精度，缺省值在 ode23 中为 tol=1E-3；在 ode45 中为 tol=1E-6 .

第 6 个输入参变量 trace 决定求解的中间结果是否显示，缺省值为 trace=0，表示不显示中间结果。

【例 6.1-1】 求方程

$$\begin{cases} y' = y - \dfrac{2x}{y}, \\ y(0) = 1 \end{cases}$$

在时间区间从 $x = 0$ 到 $x = 20$ 各节点上的数值解。

(1) 建立函数文件 ode.m

在 MATLAB 主工作窗口单击菜单 File\New\M-File，或直接点击 New M-File 工具(工具栏中第一个按钮)，进入 M 文件编辑器；然后键入：

```
function dy=ode(x,y)
dy=[y-2*x/y]
```

单击 M 文件编辑器菜单 File\Save 命令，将上述语句存储为文件名为 ode 的 M 文件。再单击 File\Close ode.m，关闭文件编辑窗口。

(2) 在 MATLAB 命令窗口中运行函数

```
[X,Y]=ode45('ode',[0 20],1)
dy =
   1
   0.9629
   ...
   3.8418e+006
```

(3) 画图观察其变化趋势

```
plot(X,Y,'-r')
axis([14 20 0 4*10^6])
```

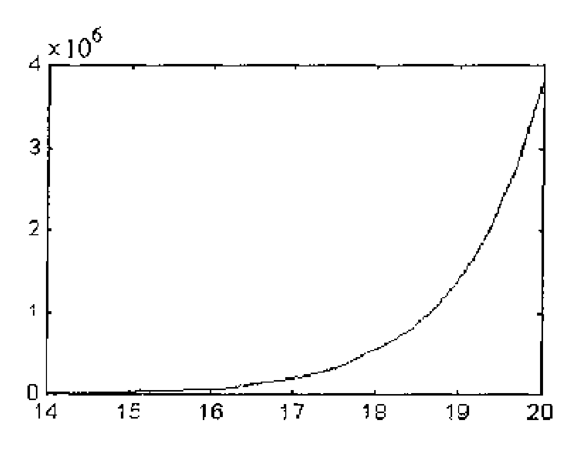


图 6-1　方程的图形解

【例 6.1-2】　求方程 $y''' + y'(y''^2 - 1) - y = 0$，初值为 $y(0)=0$，$y'(0)=1$，$y''(0)=-1$，时间区间从 $x=0$ 到 $x=20$ 各节点上的数值解。

(1) 化为标准方程

将微分方程的导数降阶，即令 $y_1 = y, y_2 = y', y_3 = y''$，则原方程变为

$$\begin{cases} y_1' = y_2, \\ y_2' = y_3, \\ y_3' = y_2(1-{y_3}^2) + y_1, \end{cases}$$

其初值条件为 $\begin{cases} y_1(0)=0, \\ y_2(0)=1, \\ y_3(0)=-1. \end{cases}$

(2) 建立函数文件 odex1.m

类似上例第 1 步的方法，新建一函数文件，命令为 odex1。函数文件 odex1.m 包含如下内容：

```
function ydot=odex1(x,y)
ydot=[y(2);y(3);y(2)*(1-y(3)^2)+y(1)];
```

(3) 解微分方程

```
[X,Y]=ode45('odex1',[0 20],[ 0 1 -1]);
```

(4) 画图观察其变化趋势

```
plot(X,Y(:,1),'-r',X,Y(:,2),':k',X,Y(:,3),'—b')
xlabel('time x');
ylabel('solution Y');
legend('Y1','Y2','Y3')
```

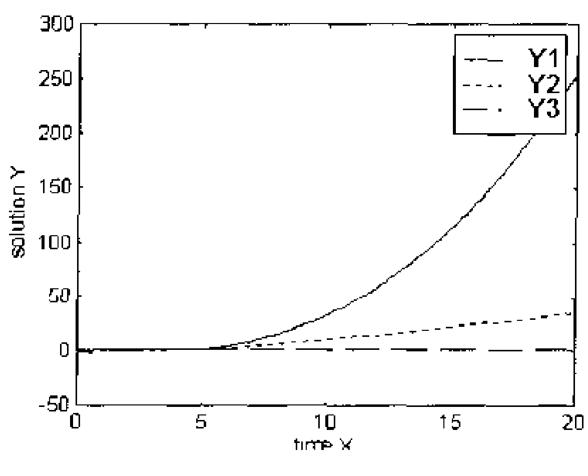


图 6-2 方程的图形解

【例 6.1-3】 求 Van der Pol 方程 $y''+(y^2-1)y'+y=0$ 的解，初值分别取 $y(0)=0$、$y'(0)=0.5$，时间区间从 $t=0$ 到 $t=20$。并绘制其解图形和相图。

(1) 将方程化为标准方程

令 $y_1=y'$，$y_2=y$，则可把方程改写成状态方程的形式：

$$\begin{cases} y_1' = y_1(1-{y_2}^2)-y_2, \\ y_2' = y_1. \end{cases}$$

(2) 建立函数文件 odex2.m

在 M 文件编辑窗口中键入下面的语句，并命名为 odex2.m 文件：

```
function ydot=odex2(t,y)   %参数必是两个(t,y)，次序不能颠倒
ydot=zeros(2,1);  %此句将零列向量赋值给ydot
ydot(1)=[(1-y(2)^2)*y(1)-y(2)];
ydot(2)=y(1);
```

(3) 解微分方程

在 MATLAB 工作窗中键入：

```
t0=[0,20];
y0=[0,0.5]';
[X,Y]=ode45('odex2',t0,y0);
plot(X,Y)
```

图形解如图 6-3 所示。

(4) 绘制相图

```
plot(Y(:,1),Y(:,2))
```

相图如图 6-4 所示。

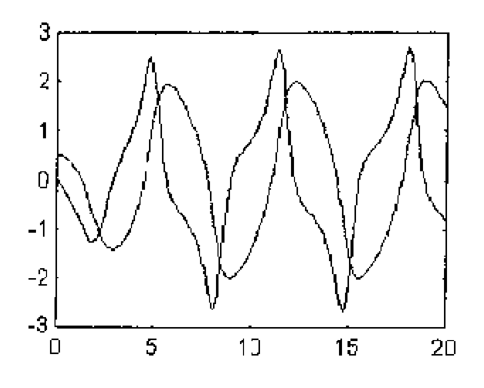



图 6-3 Van der Pol 方程的图形解

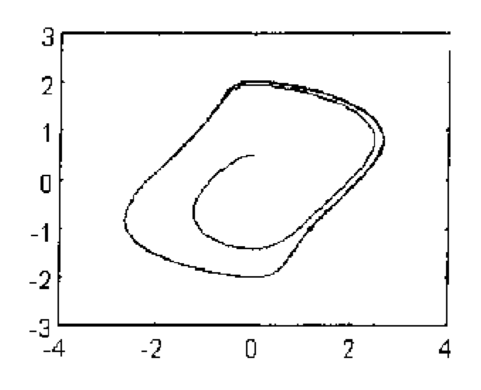


图 6-4 Van der Pol 方程的相图

刚性方程的函数命令的调用格式为

[T,Y]=ode23s('F',TSPAN,Y0)

F 是定义 $f(x, y)$ 的函数文件名。必须以列向量输出。

TSPAN=[T0,TFINAL], T0, TFINAL 以及 Y0 与非刚性方程的函数命令中的相应参数意义一致。

解向量 Y 与时间向量 T 的相应元素对应。

如果要获得指定时刻 T0,T1,…,TFINAL（增加的或减少的）的解，可输入：

```
TSPAN=[T0,T1,…,TFINAL]
```

例如，求 Van der Pol 方程的解，可输入：

```
T0=[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,...
18,19,20];
X0=[0,0.5];
[X,Y]=ode23s('odex2',T0,X0)
A=[X,Y]
```

得数值解：

```
A =

         0         0    0.5000
    1.0000   -0.6692    0.1927
    2.0000   -1.2302   -0.8595
    3.0000    0.2059   -1.4102
    4.0000    1.2303   -0.7015
    5.0000    2.2106    1.2960
    6.0000   -0.4454    1.8556
    7.0000   -1.0503    1.1196
    8.0000   -2.6431   -0.6409
```

```
 9.0000    0.1799   -1.9988
10.0000    0.8305   -1.4345
11.0000    2.0259   -0.1432
12.0000    0.6400    1.9452
13.0000   -0.6674    1.6837
14.0000   -1.4257    0.7126
15.0000   -2.1238   -1.4894
16.0000    0.4984   -1.8793
17.0000    1.0575   -1.1224
18.0000    2.6514    0.6461
19.0000   -0.1838    2.0004
20.0000   -0.8308    1.4348
```

A 中第 1 列是取不同的时刻值；第 2 列是初值为 0 时方程相应各时刻的解；第 3 列是初值为 0.5 时方程相应各时刻的解。

实际上，刚性方程的函数命令的调用格式还可以是如下的形式：

[T,Y]=ode23s('F',TSPAN,Y0,OPTION),

其中 OPTION 作为输入参数是用 odeset 函数编写的求解法则。一般此参数用来控制误差精度。采用缺省值时，相对误差(RelTol)=1E-3；绝对误差(AbsTol)=1E-6。关于 odeset 的使用方法，可参阅 MATLAB 提供的"帮助"文件。

下面是关于如何绘制常微分方程吸引子图形的两个例子。一个是绘制 Lorenz 吸引子；另一个是绘制 Chen 吸引子。

【例6.1-4】 绘制 Lorenz 吸引子。

考虑 Lorenz 系统状态方程：

$$\begin{cases} y_1'(t) = -\dfrac{8}{3}y_1(t) + y_2(t)y_3(t), \\ y_2'(t) = -10y_2(t) + 10y_3(t), \\ y_3'(t) = -y_2(t)y_1(t) + 28y_2(t) - y_3(t), \end{cases}$$

其初始条件为

$$\begin{cases} y_1(0) = 0, \\ y_2(0) = 0, \\ y_3(0) = 0.00000000001. \end{cases}$$

(1) 创建 LORENZED.m 函数文件，LORENZED.m 文件的内容是：

```
function ydot=LORENZED(t,y)
ydot=[-8/3,0,y(2);0,-10,10;-y(2),28,-1]*y;
```

(2) 在 MATLAB 命令窗口键入命令：

```
axis([10 40 -30 30 -30 30]);
view(3)
hold on
title('Lorenz Attractor')
[t,y]=ode45('LORENZED',[0,40],[0,0,0.00000000001]);
plot3(y(:,1),y(:,2),y(:,3))
```

所得图形如图 6-5 所示。



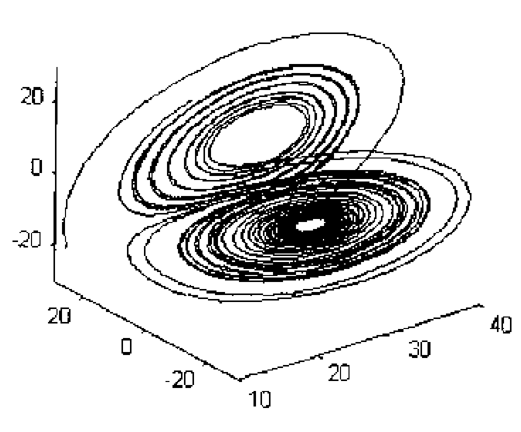


图 6-5 Lorenz 吸引子

【例 6.1-5】 美国休斯顿大学(University of Houston)的陈关荣教授在研究利用反馈控制产生混沌的问题时，发现了由一个三维系统产生的新混沌吸引子。称为 Chen's Attractor。这个三维系统是

$$\begin{cases} x' = a(y-x), \\ y' = (c-a)x - xz + cy, \\ z' = xy - bz. \end{cases}$$

下面，用 MATLAB 函数命令绘制 Chen's Attractor。取 $a = 35, b = 3, c = 28$。初始条件取为

$$\begin{cases} x(0) = -10, \\ y(0) = 0, \\ z(0) = 37. \end{cases}$$

(1) 写状态方程的 M 文件（取名为 fibno）:

```
function xdot=fibno(t,x)
xdot=[-35,35,0;-7,28,-x(1);0,x(1),-3]*x;
```

(2) 在 MATLAB 主命令窗口键入如下的命令:

```
axis([-30,30,-30,30,0,50])
view(70,10)
hold on
title('Attractor of Chen')
x0=[-10,0,37];
[t,x]=ode23('fibno',[0,30],x0);
plot3(x(:,1),x(:,2),x(:,3))
```

绘出 Chen's Attractor 图形，如图 6-6。

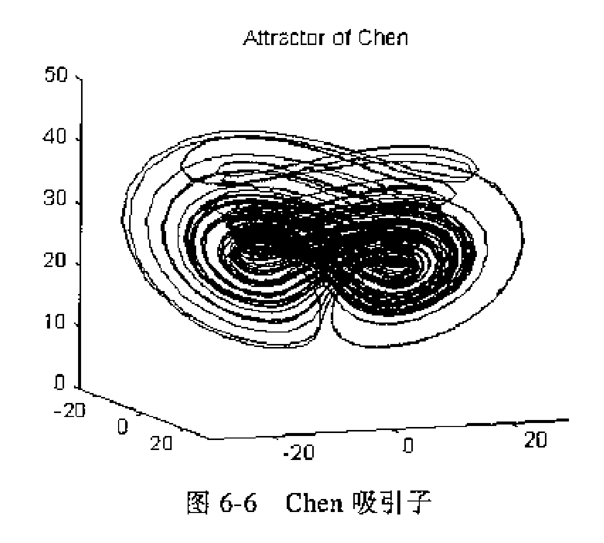


图 6-6 Chen 吸引子

6.2 求常微分方程及方程组的解析解

MATLAB 还可用于求常微分方程及常微分方程组的解析解。这需要用到 MATLAB 的符号运算工具箱中的 dsolve 命令。下面将通过实例来解释如何使用此命令。

dsolve 命令的调用格式为

dsolve('eq1','eq2','eq3',...)

其中 eqN (N=1,2,…)表示相互独立的常微分方程、初始条件或指定的自变量。如果输入的初始条件少于方程的个数，则输出结果中将出现常数 C1,C2,…等字符。关于微分方程的表达方式有如下的约定：字母 y 表示函数；Dy 表示 y 对 t 的一阶导数；Dny 表示 y 对 t 的 n 阶导数。

下面举例说明如何调用 dsolve 命令。

【例 6.2-1】　求 $\frac{dy}{dt}-\frac{2y}{x+1}=(x+1)^{\frac{5}{2}}$ 的通解。

在 MATLAB 主工作窗口中输入：

```
y=dsolve('Dy-2*y/(x+1)=(x+1)^(5/2)','x')
```

结果是

```
y =
    2/3*(x+1)^(3/2)*x^2+4/3*(x+1)^(3/2)*x+2/3*(x+1)^(3/2)
    +C1*x^2+2*C1*x+C1
```

说明　第 1 个输入变量是微分方程表达式，微分符号 D 必须大写。引号不能漏掉。第 2 个输入变量特指自变量 x。如果此项省略，解中将出现字母 t 而不是字母 x。C1 是通解中的常数。

【例 6.2-2】　求下列常系数线性齐次方程组的通解：

$$\begin{cases}\frac{dx}{dt}=3y-2z,\\ \frac{dy}{dt}=x,\\ \frac{dz}{dt}=y.\end{cases}$$

在 MATLAB 主工作窗口中键入：

```
[x,y,z]=dsolve('Dx=3y-2z','Dy=x','Dz=y')
```

结果是

```
x =
C1+t
y =
t*C1+C2+1/2*t^2
z =
1/2*t^2*C1+t*C2+C3+1/6*t^3
```

说明　dsolve 中最多只能有 12 个输入变量，但这并不妨碍解超过 12 个方程和初始条件的方程组。因为可以把多个方程定义为一个符号变量进行输入。

【例 6.2-3】　求下列初始问题的解：

$$\begin{cases}\frac{df}{dt}=3f+4g,\\ \frac{dg}{dt}=-4f+3g,\\ f(0)=0,\ g(0)=1.\end{cases}$$

在 MATLAB 主工作窗口中输入：

```
P='Df=3*f+4*g,Dg=-4*f+3*g';
V='f(0)=0,g(0)=1';
[f,g]=dsolve(P,V)
```

结果是

```
f =
exp(3*t)*sin(4*t)
g =
exp(3*t)*cos(4*t)
```

6.3 用 DEE 解常微分方程及其方程组

MATLAB 5.X 以上版本新增了微分方程编辑器（Differential Equation Editor，简记为 DEE），可以用来求常微分方程（组）的数值解，同时还可以进行动态仿真。

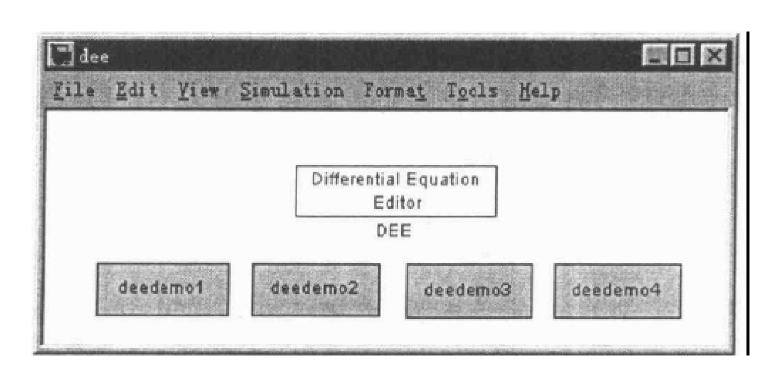
下面结合一个具体的模型介绍如何利用 MATLAB 所提供的微分方程编辑器来求常微分方程的数值解和动态仿真。

【例 6.3-1】 求 Rossler 方程组的数值解：

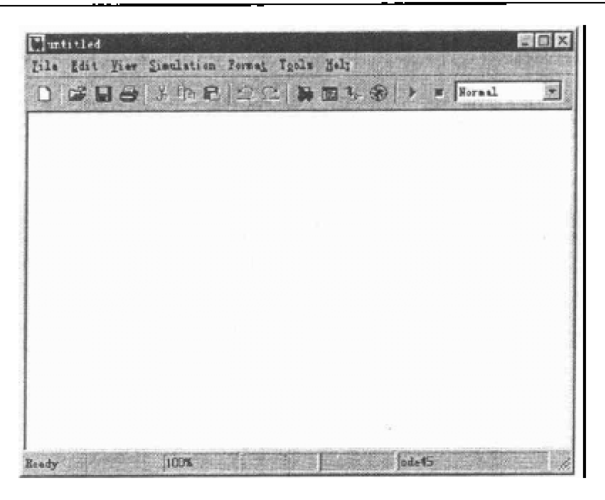
$$\begin{cases} \dfrac{dx_1(t)}{dt} = -(x_2(t) + x_3(t)), \\ \dfrac{dx_2(t)}{dt} = x_1(t) + 0.2x_2(t), \\ \dfrac{dx_3(t)}{dt} = 0.2 + x_3(x)(x_1(t) - 5). \end{cases}$$

操作步骤：

(1) 在 MATLAB 的命令窗口中键入 dee，按回车键，屏幕出现如下的界面：

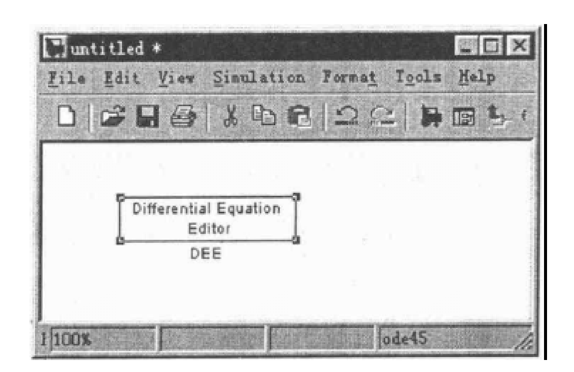


(2) 点击 File\New\Model 选项便产生一个名为 untitled 的 Simulink 编辑窗口：

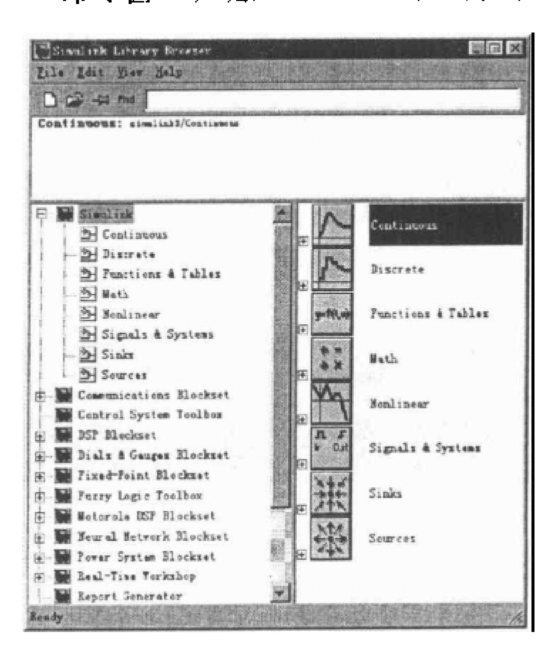


(3) 缩小此窗口，让上面的两个窗口同时显示在屏幕上。

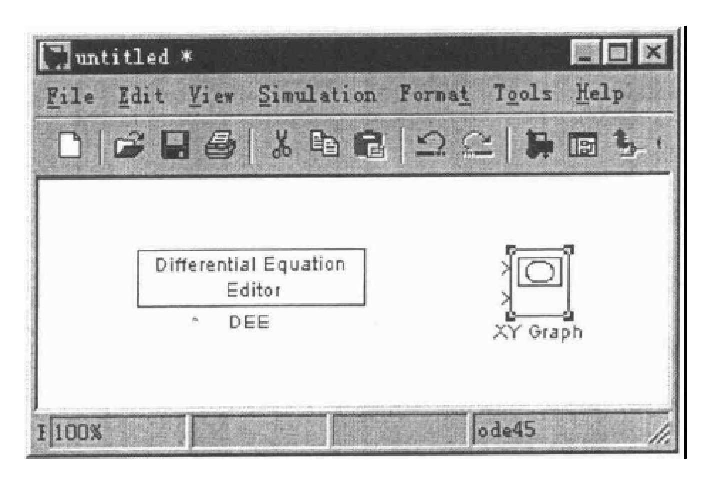
(4) 用鼠标左键点 Differential Equation Editor，并按住不放，将 DEE 拖入 untitled Simulink 工作窗口中。



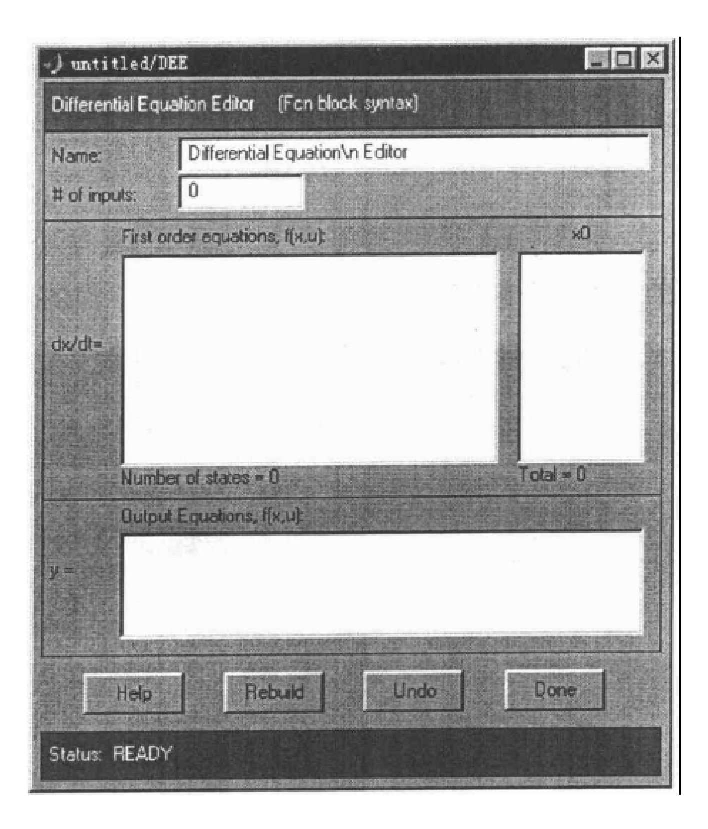
(5) 回到 MATLAB 命令窗口，键入 Simulink，出现下面的窗口：



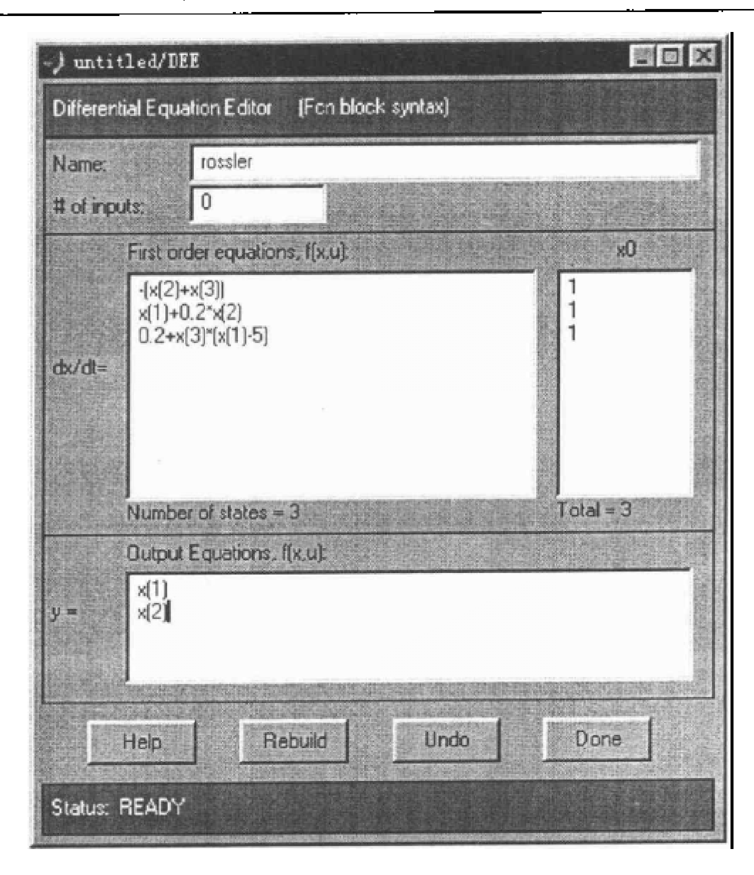
(6) 单击左栏 Simulink 目录下的 Sinks，再在其右端对应的图标中选中 XY Graph 图标，用鼠标左键按住图标不放，把它拖入 untitled Simulink 工作窗口中，如下图所示。



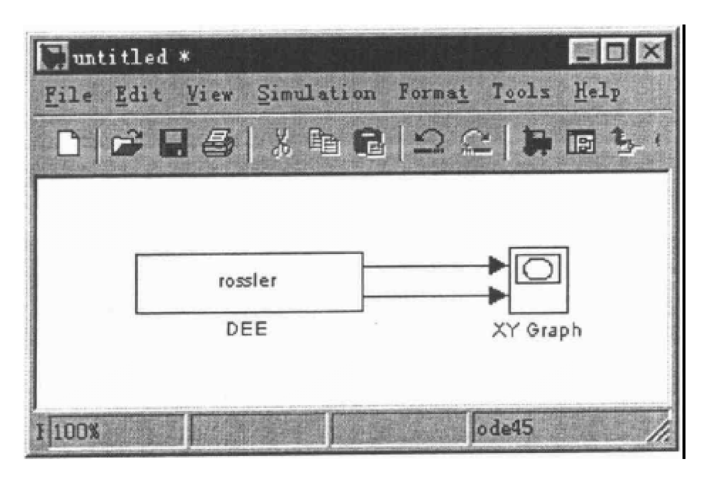
(7) 在 untitled Simulink 工作窗口中双击 Differential Equation Editor 图标，出现如下对话框：



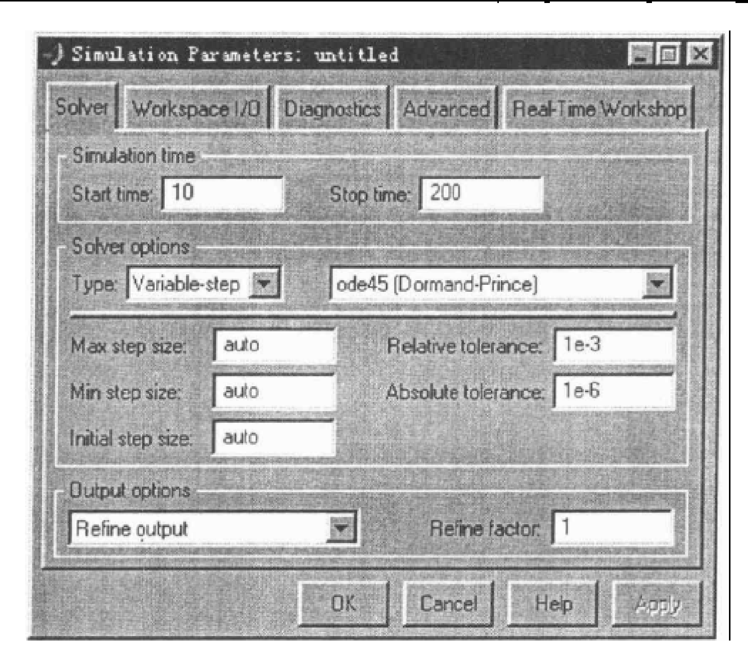
(8) 在 Name 栏中输入模型名称：rossler，dx/dt 栏中输入模型，y=栏中输入要输出的变量，x0 栏中输入初值，如下图所示。



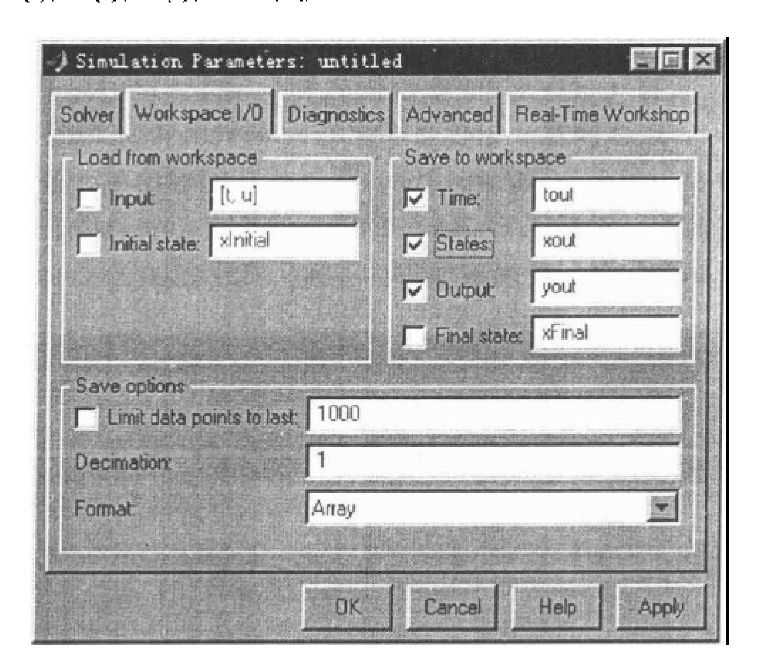
(9) 点击 Done 按钮确定，系统自动返回到 Simulink 编辑窗口，此时 DEE 模块右端出现两个小箭头，表示输出端。将光标移到输出端，待光标呈十字形时，点击鼠标并拖动，对准 XY Graph 的输入端拖曳出一直线，再松开鼠标，会出现一箭号。如法炮制，可完成模块的连接，如下图：



(10) 点选菜单 Simulation 下 Simulation parameters...选项，设置仿真参数。在打开的对话框的 Solver 标签页中，设置仿真时间段、求解器等。本例选时间段为[10,200]，用 ode45 解题。其余设置如下图所示：

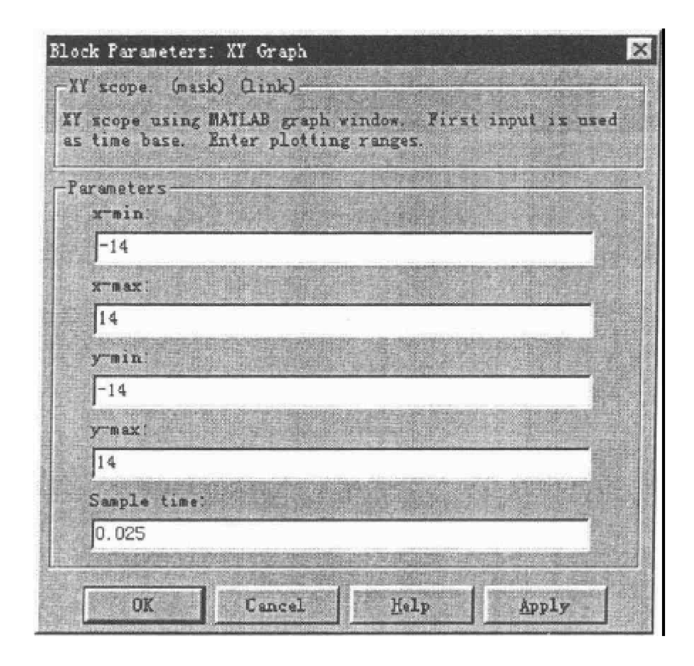


(11) 点击 Workspace I/O 标签页，设置要输出到 MATLAB 工作窗口的变量。本例欲将 t,x1(t),x2(t),x3(t)变量都输出到 MATLAB 工作窗口中，如下图所示：

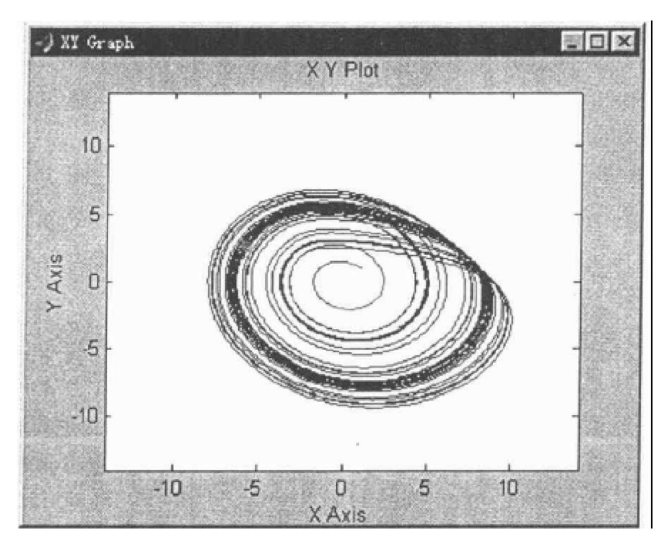


(12) 仿真参数设置完毕后，单击 Apply 按钮应用设置，单击 OK 按钮退出对话框；

(13) 最后，双击 XY Graph 模块，在打开的 Block Parameters 对话框中设置图形的显示区域和仿真步长。本例将 x 设置为从-14 到 14，y 从-14 到 14、步长为 0.025，如下图所示：



(14) 单击 Apply 按钮应用参数，再单击 OK 按钮，回到 untitled Simulink 工作窗口，选择菜单 Simulation 下 Start 命令，显示出动态仿真图形，如下图：



(15) 若要关闭，选菜单 Simulation 下的 Stop 命令。

(16) 通过 Simulation Parameters 对话框的 Solver 标签页参数的设置，可调整动态仿真图形的显示时间段。仿真结束后，在 MATLAB 工作窗口中键入命令：xout，可得数值解。此例中 xout 有 1 601 组数据，下面显示的数据仅是它的前 10 组：

```
    xout(1:10,:)
ans =
```

```
1.0000    1.0000    1.0000
0.9508    1.0295    0.9090
0.9030    1.0578    0.8258
0.8566    1.0852    0.7497
0.8112    1.1115    0.6803
0.7669    1.1369    0.6171
0.7235    1.1613    0.5595
0.6808    1.1847    0.5073
0.6389    1.2071    0.4599
0.5975    1.2287    0.4169
```

DEE 实际上是 Simulink 的一个模块，关于 Simulink 的详细使用方法，请参见参考文献[8]。

第七章　MATLAB 的基础知识

7.1　MATLAB 的启动

利用菜单、快捷键或文件夹三种方法都可进入 MATLAB 工作窗口。利用菜单的操作步骤如下：

进入 Win9x→【开始】→【程序】→【MATLAB Release 12】→【MATLAB R12】。

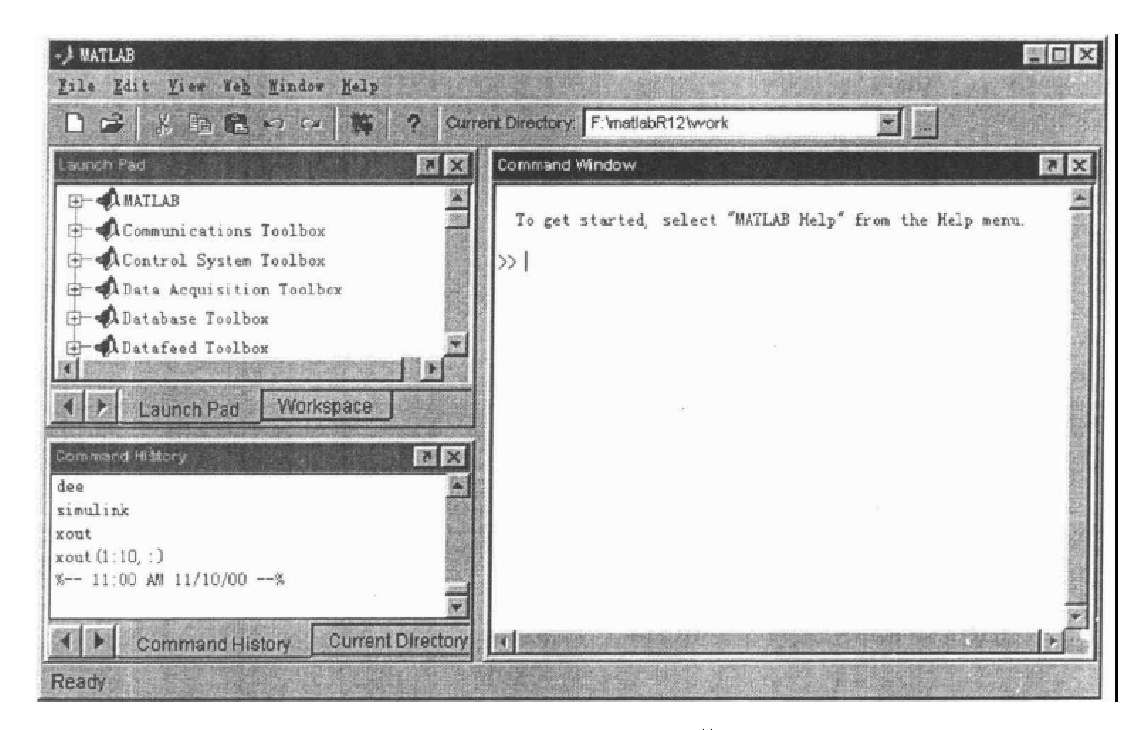


图 7-1　MATLAB 的主工作窗口

启动 MATLAB 后，系统自动运行 x:\matlabR12\toolbox\local\matlabrc.m 文件，首先按 pathdef.m 文件的要求设置系统路径，然后在工作窗口最上方显示初始提示信息，设置系统环境，运行 startup.m 文件，MATLAB 工作窗口中的第二行出现 MATLAB 环境提示符号“>>”和光标位置符。当 MATLAB

工作窗口打开后，就可以在工作窗口里进行各种运算操作。为此，本章重点介绍该工作窗口的基本环境和指令行的基本操作。MATLAB 工作窗口中的一些常用操作命令参见表 7-1。

表 7-1 MATLAB 命令窗口中部分常用命令

命令	目　的	命令	目　的
demo	演示程序	type	显示文件内容
quit	关闭和退出 MATLAB	dir	列出指定目标下的文件和子目录清单
pathtool	搜索路径管理器	whos	列出工作内存中的变量细节
path	设置 MATLAB 的搜索路径	who	列出工作内存中的变量名
lookfor	关键词检索	which	确定指定函数和文件的位置
load	从磁盘文件中调入数据变量	save	把内存数据变量存入磁盘文件中
length	确定向量的长度	workspace	工作内存浏览器
hold	控制当前图形对象是否被刷新	echo	控制运行文件指令是否显示的开关
help	在线帮助指令	what	列出当前目录下的 M 文件、mat 文件和 mex 文件
exist	检查变量或文件的存在性	size	确定矩阵的维数
clf	擦除当前图形窗口中的图形	disp	显示字符串内容
clear	从内存中清除变量和函数	pack	合并工作内存中的碎块
clc	擦除工作窗口中的显示内容	cd	改变当前工作子目录

7.2 变量、表达式和语句

变量：由字母、数字和下画线三种字符组成，但第一个字符必须为字母，且有大小写之分。

表达式：将常量或变量由算术运算符、关系运算符和逻辑运算符连接的式子。例如：

```
a=3;
b=2;
```

```
c=a>b
c =
   1
```

这里“c=”表示答案的显示，以后不再赘述。

语句：由表达式、函数以及变量、赋值号、表达式组成。例如：

```
1233/34;
sqrt(3);
a=b+c;
```

还有一些特殊变量，参见表 7-2。

表 7-2　MATLAB 的特殊变量

变量名	说　明
ans	每一次运算，没有赋值时的结果存储变量
eps	MATLAB 计算浮点数的精确度
realmax	用户电脑能表示的最大正浮点数
realmin	用户电脑能表示的最小正浮点数
pi	圆周率 π
i,j	虚数单位
Inf	无穷大 ∞
NaN	不是一个数，例如 0/0,Inf/Inf
flops	统计该工作空间中浮点数的计算次数
Version	MATLAB 版本

7.3　命令行的编辑和输入

MATLAB 采用的工作方式是一种交互式，即随时输入命令，MATLAB 可立即给出运算结果。

【例 7.3-1】　计算 $\dfrac{6\tan(0.13\pi)}{1+\sqrt{2}+\lg 5}$ 的值。

应键入：

```
6*tan(0.13*pi)/(1+sqrt(2)+log10(5))
```

然后按回车键，便可得出下面的结果：

```
ans =
     0.8340
```

若输入超过一行，可用续行符“...”。例如：

```
%计算 1 到 10 的算术和
s=1+2+3+4+5+6+7+...
8+9+10;
```

注意：MATLAB 和 C 语言相似，对变量及命令应区分大小写。这里分号“;”表示不显示运算结果。

而“%”是注释语句．该语句计算机并不执行，但通过 help 命令可把整个语句显示出来。

7.4 数据显示格式命令

MATLAB 通过 format 命令可控制数据显示格式。

【例 7.4-1】 显示 $\frac{4}{3}$ 的输出格式。

```
format long
a=4/3
a =
   1.33333333333333
```

有关 format 的具体内容参见表 7-3。

表 7-3 **数据显示格式命令 format**

命令格式	显示情况	举例说明
format 或 format short	以 5 位浮点表示	1.3333
format long	以 15 位浮点表示	1.33333333333333
format short e	以 5 位浮点指数形式表示	1.3333e+000
format long e	以 15 位浮点指数形式表示	1.333333333333333e+000
format rat	以分数形式表示	4/3
format hex	以十六进制表示	3ff5555555555555
format bank	以(金融)元、角、分表示	1.33

如果选用菜单进行数据显示格式的设置，则需点击菜单 File 下的 Preferences 选项，效果相同。

7.5 命令窗口中常用操作命令

MATLAB 提供了许多键盘输入的控制命令。

7.5.1 演示命令 demo

MATLAB 为用户提供了一套集成环境下的演示程序，图文并茂，可帮助用户尽快了解 MATLAB 的结构和功能。用户只需在命令窗口键入：

```
demo
```

回车后，在出现的对话框内选择不同的选项，便可看到 demo 的全部内容。

7.5.2 内存变量管理

1. 储存 Workspace

MATLAB 允许用户将内存中的变量，存储到一个文件里，自动形成扩展名为.mat 的文件，供以后使用。

例如，在命令窗口键入：

```
save data               %将全部内存变量存入data.mat中
save datax x            %将内存变量x存入datax.mat中
save datax x -ascii     %以ASCII格式，将内存变量x存入
                        %datax.mat中
save datay y -mat       %以二进制文件格式,将内存变量y存入
                        %datay.mat中
save datay x -append    %将内存变量x追加到datay.mat中
```

当然，也可以选择 File 菜单下 Save Workspace As...命令来进行储存。

2. 载入 Workspace

数据文件形成后，可将数据调入内存。

```
load data           %将data.mat存储的变量内容调入内存
load datay -mat     %以二进制文件格式，将datay.mat的内容
                    %调入内存
```

也可以选择 File 菜单下的 Import Data...把变量载入内存。

3. 查看 Workspace

当完成一个计算后，在内存中便会存在一些变量。我们通过命令 whos 或 who 可查看变量的情况：

```
whos
   Name      Size          Bytes   Class
   b         1×1             8     double array
   y         1×1             8     double array

Grand total is 2 elements using 16 bytes
```

选择 File 菜单下的 Import Data...命令，也可查看内存变量。

4. 删除内存中的变量

```
clear  %删除内存中全部变量
```

若选择 File 菜单下的 Import Data...命令，然后选一变量，单击 Cancel 按钮，可删除该变量。

7.5.3 搜索路径

改变路径命令 cd

如果已经在 d:\stu 下建立了一个文件夹 zhang 或目录，准备在里面工作，可将 MATLAB 主窗口中 Current Directory (当前路径)选为 d:\stu\zhang，或在命令窗口中键入：

```
cd  d:\stu\zhang
```

路径搜索

显示 MATLAB 整个系统设置的搜索路径的命令是

```
path
```

如果想设置搜索路径，可键入：

```
path(path,'d:\stu\zhang')
```

这样，MATLAB 可自动搜索你所设置的路径 d:\stu\zhang。

执行菜单 File 下的 Set Path...选项，单击 Add Folder...按钮，选择要设置为搜索路径的目录 d:\stu\zhang，按“确定”按钮，也可设置搜索路径。

7.5.4 在线帮助

若对某个函数的用法不熟悉，可使用在线帮助命令 help。

例如，查一下 sin 函数的用法：

```
help sin
```

则 MATLAB 把 sin 中注释语句显示出来：

```
SIN Sine.
   SIN(X) is the sine of the elements of X.
```

```
Overloaded methods
   help sym/sin.m
```

7.6　编程入门

7.6.1　M 文件的形式

MATLAB 有两种常用工作方式：一种是命令行操作方式；另一种是 M 文件的工作方式。

M 文件有两种类型：命令文件（Script File）和函数文件（Function File）。

命令文件

【例 7.6.1-1】　计算 $\sum_{n=1}^{20} n!$。

(1) 用记事本 Notepad 或 M 文件编辑器编写以下内容：

```
%计算 1 至 20 阶乘的和
s=0;t=1;n=1;
while n<=20
t=t*n;
s=s+t;
n=n+1;
end
s
```

(2) 选择记事本“文件”菜单中的“保存”选项，将所写内容存放于磁盘中，并起名为 prodsum.m。

在 MATLAB 主工作窗口，将 Current Directory (当前路径)选为文件 prodsum.m 存放的路径，再在 Command Window (命令窗口)中键入文件名 prodsum，按回车键确定，可显示出计算的结果：

```
s =
   2.561327494111820e+018
```

说明

① 符号“%”引导注释语句，MATLAB 不予执行，但可通过 help prodsum 命令将注释语句显示在 MATLAB 命令窗口内，供用户查询参考。

② MATLAB 程序不需要用“end”作为 M 文件的结束标志。

③ 若用户把文件 prodsum.m 存放在自己的文件夹内或工作目录上（假如路径为 d:\stu），那么在运行 prodsum.m 之前，首先应将当前路径指向 d:\stu，或将 d:\stu 置于 MATLAB 的搜索路径上，后者的方法是：在 MATLAB 命令窗

口中键入 path(path,'d:\stu')。

函数文件

【例 7.6.1-2】 把例 7.6.1-1 中的命令文件改成函数文件，并运行之。

(1) 在 M 文件编辑器或记事本中编写以下内容：

```
function f=prods(i)
% prods.m 计算1至20阶乘和的函数文件
% f=prods(i)
s=0;t=1;n=1;
while n<=i
t=t*n;
s=s+t;
n=n+1;
end
s
```

(2) 将文件存盘，取名为 prods.m。该文件定义了名为 prods 的新函数。此函数就成了 MATLAB 的宏命令，其用法与 MATLAB 中其他函数一样。

(3) 在 MATLAB 命令窗口运行以下命令：

```
prods(20)
```

这样 20 传给了程序中的 i，运行结果与例 7.6.1-1 相同。

第 1 行的 function 指明该文件是函数文件，输入参数为 i，输出参数为 f。函数名可以是 MATLAB 中任何合法的字符。输入和输出的参数类型可以是数值，也可以是字符串，参数根据实际需要设置。

函数的调用

在 MATLAB 中，函数文件可相互调用，甚至可调用它自己(即递归调用)。调用函数的格式是：

[输出参数 1,输出参数 2,……]=函数名(输入参数 1,输入参数 2,……)

【例 7.6.1-3】 计算半径为 1 至 10 不同圆的面积和周长。

(1) 建立函数文件 area.m。文件 area.m 的内容如下：

```
function [f1,f2]=area(r)
%计算圆面积和周长
f1=pi*r^2;
f2=2*pi*r;
```

(2) 编一个调用上述函数文件的命令程序，取名为 test.m。

```
R=10;  %输入圆半径
f1=zeros(1:R);f2=zeros(1:R);
```

```
for i=1:R
  [S(i),L(i)]=area(i);
end
S
L
```

(3) 在 MATLAB 命令窗口运行以下命令：

```
test
```

命令文件 test 对函数 area 调用一次，就传入参数 r 一次，由此便得到 S 和 L 的结果。

递归调用

递归调用可使程序设计方便、简洁、凝练，在程序设计中占有重要地位。

【例 7.6.1-4】　用递归调用方式计算二阶斐比拉契数列：

$$\mathrm{fib}(i)=\begin{cases}1, & i=1,2,\\ \mathrm{fib}(i-1)+\mathrm{fib}(i-2), & i>2.\end{cases}$$

(1) 编写递归调用的函数文件 fib.m。fib.m 的内容如下：

```
function f=fib(i)
% fib.m 用于计算二阶斐比拉契数列,i>0
if (i==1)|(i==2)
  f=1;
  return;
else
  f=fib(i-1)+fib(i-2);
  return;
end
```

(2) 运行函数文件 fib.m。

```
  fib(9)
ans =
    34
```

参数传递

在函数文件中,可检查传入或传出的参数数目,这样可使编程具有很大的灵活性。其函数如下：

nargin	检查在调用该函数文件时传入参数数目。
nargout	检查在调用该函数文件时传出参数数目。

【例 7.6.1-5】　以函数 plot.m 的意图说明 nargin 的用法（该函数不带输出参数，但有 3 个可选的输入参数）。

```
function test(x,y,x1,y1)
  if nargin==0,
    disp('??? Error using==>plot')
    disp('Not enough input arguments')
  elseif nargin==1  %有1个输入参数即nargin=1
    plot(x);
  elseif nargin==2  %有2个输入参数即nargin=2
    plot(x,y);
  elseif nargin==3   %有3个输入参数即nargin=3
    plot(x,y,x1);
  elseif nargin==4  %有4个输入参数即nargin=4
    plot(x,y,x1,y1);
end
```

7.6.2 控制语句

MATLAB 语言和其他计算机语言一样，也有程序结构：(1) 顺序结构；(2) 循环结构；(3) 分支结构。

一、循环结构

MATLAB 语言提供了两种循环方式：for-end 循环和 while-end 循环。

for-end 循环

与 BASIC，C，FORTRAN，PASCAL 等其他计算机语言一样，循环条件是有规律地变化的，通常是把循环条件的初值、表达式放在循环的开头，这种形式就是 for 循环结构。for-end 循环的一般格式是：

```
for i=〈表达式〉
  〈语句体〉
end
```

通常，〈表达式〉是一个向量，如 m:s:n.s 给出循环的范围和步长。该向量的元素被逐一赋值给循环变量 i，然后执行〈语句体〉。

【例 7.6.2-1】 简单的循环。

```
for i=1:5
  x(i)= sin(i);
end
x
x =
  0.8415    0.9093    0.1411   -0.7568   -0.9589
```

说明 在 i=1:5 这个语句中，1 是循环初值，5 是终值，缺省步长为 1。〈循

环体〉实现对 x 的 5 个元素赋值。循环还可以嵌套，但需注意，第一个 for 必须与 end 相匹配，否则程序会出错。

while-end 循环

While 循环使〈语句体〉在逻辑条件控制下重复执行若干次，直到循环条件为假时终止。While 循环的一般格式是：

While〈表达式〉
　〈语句体〉
end

只要〈表达式〉条件满足，〈语句体〉就重复执行。当〈表达式〉结果不是标量时，可以用 any, all 等函数处理。如果表达式为 1 时，该循环将无限制地进行下去。

【例 7.6.2-2】 利用例 7.6.1-2 的阶乘函数 prods，计算 1 到 20 阶乘之和。

```
s=0;n=1;
while n<=20
s=s+prods(1:n);
n=n+1;
end
s
s =
   2.5613e+018
```

二、选择结构

if-end 选择结构

选择结构的一般形式是：

if 〈逻辑表达式〉
　〈语句体 1〉
else
　〈语句体 2〉
end

else 部分表示当〈逻辑表达式〉结果为假时，就执行 else 后面的〈语句体 2〉。

【例 7.6.2-3】 比较'a'和'b'的大小。

```
a='a';b='b';
if (a<b)
  disp('true')
else
```

```
    disp('false')
  end
```

if 语句嵌套

```
if 〈逻辑表达式 1〉
   〈语句体 1〉
elseif 〈逻辑表达式 2〉
   〈语句体 2〉
……
else
   〈语句体 else〉
end
```

需要注意：if 语句嵌套时，if 和 else 必须对应，否则容易出错。在 else 子句中也可嵌套 if 语句，这就形成了 elseif 结构。

【例 7.6.2-4】　选择灰色等级(如图 7-2)。

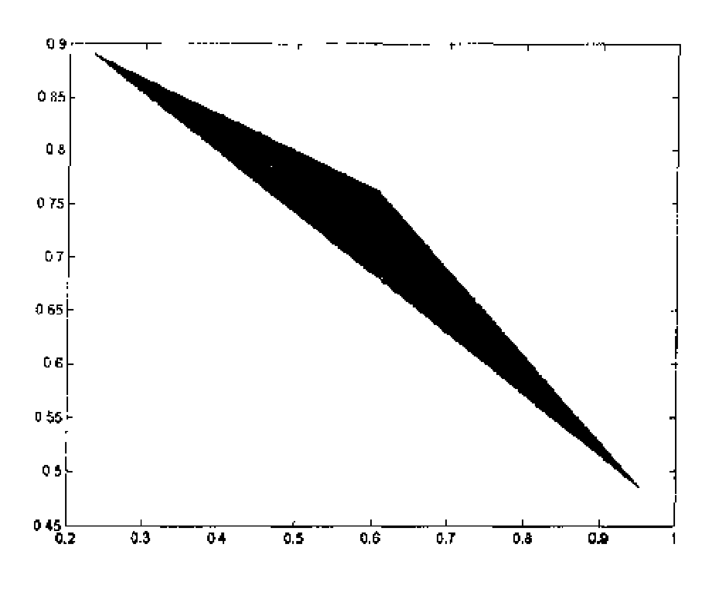


图 7-2　灰色等级选择

(1) 编写函数文件 incolor.m。incolor.m 的内容如下：

```
function gd=incolor(colr)
  gd=[0 0 0];
if colr<3;
  gd=[.8  .8  .8];
elseif (colr>=3)&(colr<6);
  gd=[.6  .6  .6];
```

```
else
  gd=[.4  .4  .4];
end
```

(2) 在 MATLAB 命令窗口中运行：

```
incolor(9)
fill(rand(1,3),rand(1,3),ans);
```

运行结果如图 7-2 所示。

switch 多分支选择结构

switch 语句和 if 语句类似，可根据变量或表达式的不同取值分别执行不同的命令。其基本调用格式为：

```
switch 〈表达式〉
case data 1
   〈语句体 1〉
case data 2
   〈语句体 2〉
……
otherwise
   〈语句体〉
end
```

【例 7.6.2-5】 用菜单选择灰色等级。

```
echo off
gd=[0  0  0];
m=menu('请选择灰色等级','浅灰','中灰','深灰')   % menu可对每
    %行菜单项返回自然数
switch m
case 1
  gd=[.8  .8  .8];
case 2
  gd=[.6  .6  .6];
otherwise
  gd=[.4  .4  .4];
end
fill(rand(1,3),rand(1,3),gd);
```

命令运行后，出现 MENU 对话框，有浅灰、中灰、深灰三种灰色等级选择，选择后出现对应的灰色等级图示。

附录一 MATLAB 的函数命令

常用指令

管理指令和函数

demo	演示程序
help	帮助命令
lookfor	关键词检索
path	控制 MATLAB 的搜索路径
pathtool	路径管理器
type	显示文件内容
what	列出当前目录上的 M 文件、mat 文件和 mex 文件
which	确定指定函数和文件的位置

变量和内存空间的管理

clear	从内存中清除变量和函数
disp	显示矩阵和文字内容
length	确定向量的长度
load	从磁盘中调入数据变量
pack	合并工作内存中的碎块
save	把内存变量存入磁盘
size	确定矩阵的维数
who	列出工作内存中的变量名
whos	列出工作内存中的变量细节
workspace	工作内存浏览器

调用操作系统和文件处理

cd	改变当前工作目录

delete	删除文件
diary	储存 MATLAB 指令窗口操作内容
dir	列出文件
getenv	给出环境值
!	执行外部应用程序

指令窗控制

clc	清除窗口命令
format	设置数据输出格式
home	光标返回行首
echo	显示命令文件指令的切换开关
more	命令窗口分页输出的控制开关

MATLAB 的启动和终止

quit	退出 MATLAB
matlabrc	MATLAB 的主启动文件
startup	启动 MATLAB 时的自动执行 M 文件

通用信息查询

hostid	MATLAB 服务中心识别号
whatsnew	未入档的新特征信息
info	关于 MATLAB 和 Math Works 公司的信息
subscribe	MATLAB 用户注册
ver	MATLAB, Simulink 和 Toolbox 的版本信息

基本平面图形

fill	平面多边形填色
loglog	双对数刻度曲线
plot	直角坐标下线性刻度曲线
semilogxX	轴半对数刻度曲线
semilogyY	轴半对数刻度曲线

特殊平面图形

bar	直方图
compass	从原点出发的复数向量图

comet	彗星状轨迹图
errorbar	误差棒棒图
feather	从 X 轴出发的复数向量图
fplot	函数曲线图
hist	统计频数直方图
polar	极坐标曲线图
rose	统计频数扇形图
stairs	阶梯形曲线图
stem	火柴杆图

二维图形注释

grid	画坐标网格线
gtext	用鼠标在图上标注文字
legend	图例说明
text	在图上标注文字
title	图形标题
xlabel	X 轴名标注
ylabel	Y 轴名标注

线、面填色指令

comet3	三维彗星动态轨迹线图
fill3	三维曲面多边形填色
plot3	三维直角坐标曲线图

三维数据的等高线和其他二维表现

clabel	给等值线加标注
contour	等值线图
contourc	等值线计算
contour3	三维等值线图
quiver	矢量场图

曲面与网线图

mesh	三维网线图
meshc	带等值线的三维网线图
meshz	带零基准面的三维网线图

surfc	带等值线的三维表面图
surfl	带光照的三维表面渲染图
surf	三维表面图

内剖视图

slice	切片图

图的表现

axis	轴的刻度和表现
caxis	(伪)颜色轴刻度
colormap	设置彩色图
hidden	消隐
shading	图形渲染模式
view	设定 3-D 图形观测点
viewmtx	观测点转换矩阵

三维图的注释

grid	画坐标网格线
gtext	用鼠标在图上标注文字
text	在图上标注文字
title	图形标题
xlabel	X 轴名标注
zlabel	Z 轴名标注

其他三维图形对象

cylinder	圆柱面
sphere	球面

基本数学函数

三角函数

sin	正弦函数
sinh	双曲正弦函数
asin	反正弦函数

asinh	反双曲正弦函数
cos	余弦函数
cosh	双曲余弦函数
acos	反余弦函数
acosh	反双曲余弦函数
tan	正切函数
tanh	双曲正切函数
atan	反正切函数
atan2	反正切主值
atanh	反双曲正切
sec	正割函数
sech	双曲正割函数
asec	反正割函数
asech	反双曲正割函数
csc	余割函数
csch	双曲余割函数
acsc	反余割函数
acsch	反双曲余割函数
cot	余切函数
coth	双曲余切函数
acot	反余切函数
acoth	反双曲余切函数

指数函数

exp	以 e 为底的指数函数
log	自然对数函数
log10	常用对数函数
log2	以 2 为底的对数并分解浮点数
pow2	底为 2 的幂并标出浮点数
sqrt	开平方
nextpow2	高于 2 次幂的函数

复函数

abs	取模
angle	相角

conj	共轭复数
imag	虚部
real	实部

专用变量和常数

ans	当前答案
eps	浮点相对精度
realmax	最大正浮点数
realmin	最小正浮点数
pi	3.1415926535897···
i, j	虚数单位
Inf	无穷大
NaN	非数
isnan	非数为真

关于函数的命令和 ODE 求解器

优化和求根

fmin	求单变量的最小函数
fmins	求多变量的最小函数
fzero	求单变量函数的零点

数值积分

quad	用低阶法求数值积分
quad8	用高阶法求数值积分
dblquad	求二重数值积分

绘图命令

ezplot	简易函数绘图命令
fplot	绘图函数

常微分方程求解器

(若不能断定是否为刚性方程，则先使用 ode45，然后再用 ode15s)

ode45	用较高阶法解非刚性微分方程

ode23	用低阶法解非刚性微分方程
ode113	用变阶法解非刚性微分方程
ode23t	用梯形法则解刚性微分方程
ode15s	用变阶法解刚性微分方程
ode23s	用低阶法解刚性微分方程
ode23tb	用低阶法解刚性微分方程
odefile	ODE 文件句法

ODE 选项操作

odeset	创建/改变 ODE 选项结构
odeget	获得 ODE 选项参数

ODE 输出函数

odeplot	时间序列 ODE 输出函数
odephas2	ODE 2 维相平面输出函数
odephas3	ODE 3 维相平面输出函数
odeprint	ODE 打印命令窗口

演示程序

deedemo1	Van der Pol 方程
deedemo2	Lorenz 吸引子
deedemo3	质量和弹簧
deedemo4	质量和弹簧系统动画

偏微分方程工具箱

PDE 算法

adaptmesh	生成自适应网格和 PDE 的解
assema	组装面积积分贡献
assemb	组装边界条件贡献
assempde	组装一个 PDE
hyperbolic	解双曲型问题
parabolic	解抛物型问题
pdeeig	解特征值 PDE 问题

pdenonlin	解非线性 PDE 问题
poisolv	求矩形网格上 Poisson 方程的快速解

用户界面算法和功能

pdecirc	画圆
pdeellip	画椭圆
pdemdlcv	将 MATLAB 4.2c 的 M-文件转为能为 MATLAB 5 所用
pdepoly	画多边形
pderect	画矩形
pdetool	PDE Toolbox 的用户界面（GUI）

几何算法

csgchk	检查几何描述矩阵的有效性
csgdel	删除最小区域之间的边界
decsg	分解几何结构成最小区域
initmesh	建立初始三角形网格
jigglemesh	优化三角形网格
pdearcl	弧长和参数表达之间内插值
poimesh	矩形网格上建立规则网格
refinemesh	加密三角形网格
wbound	写边界条件的详细数据文件
wgeom	写几何区域的详细数据文件

绘图函数

pdecont	绘制轮廓图的速记命令
pdegplot	绘 PDE 几何图
pdemesh	绘 PDE 三角形网格图
pdeplot	PDE Toolbox 的一般绘图函数
pdesurf	绘制曲面图的速记命令

功能算法

dst	离散正弦变换
idst	离散正弦逆变换
pdeadgsc	用相对误差准则选择低劣三角形
pdeadworst	用最低劣值选择三角形

pdecgrad	计算 PDE 解的通量
pdeent	与已知三角形（集）相邻三角形的索引
pdegrad	计算 PDE 解的梯度
pdeintrp	插函数值于三角形中心处
pdejmps	为适应变化而进行误差估计
pdeprtni	于网格节点处插函数值
pdesde	与子区域集相邻的边界索引
pdesdp	子区域内（节）点的索引
pdesdt	子区域内三角形的索引
pdesmech	计算结构力学张量函数
pdetrg	三角形几何数据
pdetriq	测量三角形网格质量
poiasma	Poisson 方程边界点矩阵贡献
poicalc	矩形网格上 Poisson 方程的快速解
poiindex	矩形网格正则次序点的索引
sptarn	稀疏特征问题生成的解
tri2grid	将 PDE 三角形网格转化为矩形网格

用户自定义算法

pdebound	边界 M-文件
pdegeom	几何 M-文件

演示程序

pdedemo1	单位圆盘上 Poisson 方程的精确解
pdedemo2	解 Helmholtz 方程和研究反射波
pdedemo3	解最小表面问题
pdedemo4	用子区域分裂解 PDE 问题
pdedemo5	解抛物型方程（热传导方程）
pdedemo6	解双曲型方程（波动方程）
pdedemo7	点源自适应问题
pdedemo8	矩形网格上解 Poisson 方程

附录二　根据有限元法用 MATLAB 语言解 PDE 的程序

解同轴电缆问题（5.6 节）的程序：

```
clear
  %这段程序处理区域为G1，且进行节点编号、单元编号。节点和单元编号都是按
  %由下到上由左到右的规则。然后利用图论绘图方式绘制G1的网格图。
xmin1=input('请输入区域左端的x值');
xmax1=input('请输入区域右端的x值');
ymin1=input('请输入区域下端的y值');
ymax1=input('请输入区域上端的y值');
n1=input('请输入区域划分的纵线数');
m1=input('请输入区域划分的横线数');
a1=zeros(m1*n1);b1=zeros(m1*n1);
xy1=zeros(m1*n1,2);x1=zeros(1,m1*n1);y1=zeros(1,m1*n1);
%计算节点坐标
for k=1:n1
  for i=1:m1
    x1(i+(k-1)*m1)=xmin1+(k-1)*(xmax1-xmin1)/(n1-1);
    y1(i+(k-1)*m1)=ymin1+(i-1)*(ymax1-ymin1)/(m1-1);
  end
end
clear k i
a1(m1,m1)=2;
a1((n1-1)*m1+1,(n1-1)*m1+1)=2;
a1(1,1)=3;
a1(n1*m1,n1*m1)=3;
for i=1:(n1-2)
  a1(i*m1+1,i*m1+1)=4;
  a1((i+1)*m1,(i+1)*m1)=4;
end
clear i
for j=2:(m1-1)
```

```
  a1(j,j)=4;
  a1((n1-1)*m1+j,(n1-1)*m1+j)=4;
end
clear j
for i=1:(n1-2)
  for j=2:(m1-1)
  a1(i*m1+j,i*m1+j)=6;
  end
end
clear i j
a1=sparse(a1);
clear i j
for i=1:n1*m1-1
  b1(i,i+1)=-1;
end
clear i
for i=1:(n1-1)*m1
  b1(i,i+m1)=-1;
end
clear i
for i=1:(n1-1)*m1-1
  b1(i,i+m1+1)=-1;
end
for k=1:(n1-1)
for h=1:(n1-1)
    b1(m1*k,m1*h+1)=0;
  end
end
clear k h
b1=sparse(b1);
b1=a1+b1'+b1;
clear a1
a1=sparse(b1);
clear b1
  xy1=[x1',y1'];
subplot(2,2,1);
gplot(a1,xy1);  %绘制网格图
e1=zeros(2,3);
u1=[1,1,1];
```

```
 for k=1:(n1-2)
   for i=1:(m1-1)
     e1=[i+(k-1)*m1,i+k*m1,k*m1+1+i;...
     i+(k-1)*m1,k*m1+1+i,i+(k-1)*m1+1];
     u1=[u1;e1];
   end
 end
 clear k i e1
 for i=1:(m1-1)
 e1=[i+(n1-2)*m1,i+(n1-1)*m1+(m1-1)/2,...
 (n1-1)*m1+1+i+(m1-1)/2;i+(n1-2)*m1,...
 (n1-1)*m1+1+i+(m1-1)/2,i+(n1-2)*m1+1];
   u1=[u1;e1];
 end
 u1(1,:)=[];
 clear k i e1
 %这段程序处理区域G2。功能同上。
 xmin2=input('请输入区域左端的x值');
 xmax2=input('请输入区域右端的x值');
 ymin2=input('请输入区域下端的y值');
 ymax2=input('请输入区域上端的y值');
 n2=input('请输入区域划分的纵向线数');
 m2=input('请输入区域划分的横向线数');
 a2=zeros(m2*n2);b2=zeros(m2*n2);
 xy2=zeros(m2*n2,2);x2=zeros(1,m2*n2);y2=zeros(1,m2*n2);
 % calculation node axit
 for k=1:n2
   for i=1:m2
     x2(i+(k-1)*m2)=xmin2+(k-1)*(xmax2-xmin2)/(n2-1);
     y2(i+(k-1)*m2)=ymin2+(i-1)*(ymax2-ymin2)/(m2-1);
   end
 end
 clear k i
 a2(m2,m2)=2;a2((n2-1)*m2+1,(n2-1)*m2+1)=2;a2(1,1)=3;
 a1(n2*m2,n2*m2)=3;
 for i=1:(n2-2)
   a2(i*m2+1,i*m2+1)=4;
   a2((i+1)*m2,(i+1)*m2)=4;
 end
```

```
clear i
for j=2:(m2-1)
  a2(j,j)=4;
  a2((n2-1)*m2+j,(n2-1)*m2+j)=4;
end
clear j
for i=1:(n2-2)
  for j=2:(m2-1)
  a2(i*m2+j,i*m2+j)=6;
  end
end
clear i j
a2=sparse(a2);
clear i j
for i=1:n2*m2-1
  b2(i,i+1)=-1;
end
clear i
for i=1:(n2-1)*m2
  b2(i,i+m2)=-1;
end
clear i
for i=1:(n2-1)*m2-1
  b2(i,i+m2+1)=-1;
end
for k=1:(n2-1)
for h=1:(n2-1)
    b2(m2*k,m2*h+1)=0;
  end
end
clear k h
b2=sparse(b2);
b2=a2+b2'+b2;
clear a2
a2=sparse(b2);
clear b2
  xy2=[x2',y2'];
hold on;
subplot(2,2,1);
```

```
gplot(a2,xy2);
e2=zeros(2,3);
u2=[0,0,0];
for k=1:(n2-1)
  for i=1:(m2-1)
    e2=[i+(k-1)*m2+m1*(n1-1),i+k*m2+m1*(n1-1),...
    k*m2+1+i+m1*(n1-1);i+(k-1)*m2+m1*(n1-1),...
    k*m2+1+i+m1*(n1-1),i+(k-1)*m2+1+m1*(n1-1)];
    u2=[u2;e2];
  end
end
u2(1,:)=[];
clear k i e2
x=[x1(1,1:m1*(n1-1)),x2(1,1:m2*n2)];
 y=[y1(1,1:m1*(n1-1)),y2(1,1:m2*n2)];
 u=[u1;u2];
%计算刚度矩阵[K]、节点数和节点坐标
N=m2*(n1+n2-1)-(n1-1)*(m1-1)/2;
K=zeros(N);
%开始计算矩阵[K]
for h=1:2*(m1-1)*(n1-1)+2*(m2-1)*(n2-1)
 mi=1;mj=2;mk=3;
 i=u(h,1);j=u(h,2);k=u(h,3);
 sm=[1,x(i),y(i);1,x(j),y(j);1,x(k),y(k)];
s(h)=det(sm)/2;
%计算Kij
 Kii=((y(j)-y(k))^2+(x(k)-x(j))^2)/(4*s(h));
 K(i,i)=K(i,i)+Kii;
 Kjj=((y(k)-y(i))^2+(x(k)-x(i))^2)/(4*s(h));
 K(j,j)=K(j,j)+Kjj;
 Kkk=((y(j)-y(i))^2+(x(j)-x(i))^2)/(4*s(h));
 K(k,k)=K(k,k)+Kkk;
       Kij=((y(j)-y(k))*(y(k)-y(i))+(x(k)-x(j))*(x(i)-x(k)))/(4*s(h));
 K(i,j)=K(i,j)+Kij;
 K(j,i)=K(j,i)+Kij;
       Kjk=((y(k)-y(i))*(y(i)-y(j))+(x(i)-x(k))*(x(j)-x(i)))/(4*s(h));
 K(j,k)=K(j,k)+Kjk;
 K(k,j)=K(k,j)+Kjk;
       Kki=((y(i)-y(j))*(y(j)-y(k))+(x(j)-x(i))*(x(k)-x(j)))/(4*s(h));
```

```
   K(k,i)=K(k,i)+Kki;
   K(i,k)=K(i,k)+Kki;
 end
 v=sparse(K);
 clear i j k h mi mj mk sm  Kii Kjj Kkk Kij Kjk Kki K
 %解线性方程组
  for i=0:m2
    v(N-i,:)=[];
  end
  clear i
  for i=0:(n2-3)
    v(N-(2+i)*m2,:)=[];
  end
  clear i
  for i=1:(m1+1)/2
    v(N-(n2-1)*m2-(m1-2)-i,:)=[];
  end
  clear i
  for i=0:(n1-2)
v(N-n2*m2-i*m1,:)=[];
v(N-n2*m2+1-(1+i)*m1,:)=[];
end
clear i
q=v(:,1);
for i=0:(n1-2)
  q1=v(:,1+m1*i);q2=v(:,m1*(1+i));
  q=[q,q1,q2];
end
q(:,1)=[];
clear i;
for i=1:(m1+1)/2
q1=v(:,m1*(n1-1)+i);q=[q,q1];
end
clear i
for i=0:(n2-2)
  q1=v(:,m1*(n1-1)+m2*(i+1));q=[q,q1];
end
clear i
for i=(N-m2+1):N
```

```
   q1=v(:,i);q=[q,q1];
  end
  for i=0:m2
   v(:,N-i)=[];
  end
  clear i
  for i=0:(n2-3)
   v(:,N-(2+i)*m2)=[];
  end
  clear i
  for i=1:(m1+1)/2
   v(:,N-(n2-1)*m2-(m1-2)-i)=[];
  end
  clear i
  for i=0:(n1-2)
   v(:,N-n2*m2-i*m1)=[];
   v(:,N-n2*m2+1-(1+i)*m1)=[];
  end
  clear i
  M1=N-2*n1-n2-m2-(m1+1)/2+3;
  f=ones(M1,1);f=4*s(1)*f;
   for i=1:(m1-2)
     f(i,1)=2*s(1);
   end
  clear i
   for i=0:(n2-3)
     f((m1-2)*n1+i*(m2-1)+1,1)=2*s(1);
   end
  clear i
  l=1;
   for i=1:(n1-1)
  l=[l,0];l=[l,10];
   end
  l(1)=[];
  clear i
   for i=1:(m1+1)/2
  l=[l,0];
   end
  clear i
```

```
  for i=1:(n2-1)+m2
 l=[1,10];
  end
 clear i
 g=f-q*l';r=v\g;
 %后处理工作
 w3=zeros(m1-2,1);
 for i=1:n1
   w1(i)=0;w2(i)=10;
   w3=[w3,r((i-1)*(m1-2)+1:i*(m1-2),1)];
 end
 w3(:,1)=[];
 w=[w1;w3;w2];
 clear i
 w4=0;
 for i=1:(m1+1)/2-1
   w4=[w4;0];
 end
 clear i
 w4=[w4;r((m1-2)*(n1-1)+1:(m1-2)*n1,1)];
 for i=0:(n2-3)
   w4=[w4,r((m1-2)*n1+(m2-1)*i+1:(m1-2)*n1+(m2-1)*(i+1),1)];
 end
 clear i
 w5=10*ones((m2-1),1);
 w6=[w4,w5];w7=10*ones(1,n2);w6=[w6,w7];
 t1=xmin1:(xmax1-xmin1)/(n1-1):xmax1;
 T1=ymin1:(ymax1-ymin1)/(m1-1):ymax1;
 t2=xmin2:(xmax2-xmin2)/(n2-1):xmax2;
 T2=ymin2:(ymax2-ymin2)/(m2-1):ymax2;
 subplot(2,2,2);meshc(t1,T1,w);axis([0 2 0 3 0 10]);hold on;
 meshc(t2,T2,w6);
 [t1,T1]=meshgrid(t1,T1);
 [t2,T2]=meshgrid(t2,T2);
 subplot(2,2,3);surfc(t1,T1,w);axis([0 2 0 3 0 10]);
 hold on; surfc(t2,T2,w6);view(60,20);axis([0 2 0 3 0 10]);
 subplot(2,2,4);contour(t1,T1,w);axis([0 2 0 3 ]);
 hold on; contour(t2,T2,w6);axis([0 2 0 3 ]);
 view(0,90);
 w1=flipud(w),w2=flipud(w6)
```

参 考 文 献

[1] 姜礼尚，庞之垣编著. 有限元方法及其理论基础. 北京：人民教育出版社，1979
[2] 李荣华，冯果忱编著. 李荣华修订. 微分方程数值解法（第三版）. 北京：高等教育出版社，1996
[3] 张宜华编著. 史惠康审校. 精通 MATLAB 5. 北京：清华大学出版社，1999
[4] The MathWorks. Partial Differential Equation Toolbox For Use with MATLAB, 1997
[5] 施阳等编著. MATLAB 语言精要及计算仿真工具 SIMULINK. 北京：西北工业大学出版社，1999
[6] 施阳，李俊等编著. MATLAB 语言工具箱——TOOLBOX 实用指南. 北京：西北工业大学出版社，1999
[7] 楼顺天，于卫，阎华梁编著. MATLAB 程序设计语言. 西安：西北电子科技大学出版社，1997
[8] 尚涛等编著. 工程计算可视化的方法与 MATLAB 实现. 武汉：武汉大学出版社，2001